New Techniques for Arabic Document Classification

Hamouda Khalifa Hamouda Chantar

Submitted in fulfilment of the requirements

of the degree of Doctor of Philosophy

Heriot-Watt University

School of Mathematical and Computer Sciences

September 2013

Abstract

Text classification (TC) concerns automatically assigning a class (category) label to a text document, and has increasingly many applications, particularly in the domain of organizing, for browsing in large document collections. It is typically achieved via machine learning, where a model is built on the basis of a typically large collection of document features. Feature selection is critical in this process, since there are typically several thousand potential features (distinct words or terms). In text classification, feature selection aims to improve the computational efficiency and classification accuracy by removing irrelevant and redundant terms (features), while retaining features (words) that contain sufficient information that help with the classification task.

This thesis proposes binary particle swarm optimization (BPSO) hybridized with either K Nearest Neighbour (KNN) or Support Vector Machines (SVM) for feature selection in Arabic text classification tasks. Comparison between feature selection approaches is done on the basis of using the selected features in conjunction with SVM, Decision Trees (C4.5), and Naive Bayes (NB), to classify a hold out test set. Using publically available Arabic datasets, results show that BPSO/KNN and BPSO/SVM techniques are promising in this domain. The sets of selected features (words) are also analyzed to consider the differences between the types of features that BPSO/KNN and BPSO/SVM tend to choose. This leads to speculation concerning the appropriate feature selection strategy, based on the relationship between the classes in the document categorization task at hand.

The thesis also investigates the use of statistically extracted phrases of length two as terms in Arabic text classification. In comparison with Bag of Words text representation, results show that using phrases alone as terms in Arabic TC task decreases the classification accuracy of Arabic TC classifiers significantly while combining bag of words and phrase based representations may increase the classification accuracy of the SVM classifier slightly.

Acknowledgements

I would like to thank my supervisor Prof David Corne for his continued encourgement and guidance in every step of my PhD. He always finds time to discuss the research and give advice.

Contents

List of Figures

List of Tables

Acronyms

ACO ANT Colony Optimization

ANN Artificial Neural Networks

ARFF Attribute Relation File Format

AI Artificial Intelligence

BPSO Binary Particle Swarm Optimization

BOW Bag Of Words

C4.5 Decision Trees Classifier

CC Correlation Coefficient

CHI Chi Square Statistics

DF Document Frequency

FN False negatives

FP False Positives

FS Feature Selection

FSS Feature Subset Selection

FIPSO Fully informed particle swarm optimization

FCM Fuzzy C-Means

IDF Inverse Document Frequency

IG Information Gain

KE Knowledege Engineering

KNN K Nearest Neighbour

LDA Linear Discriminant Analysis

LOOCV Leave One Out Cross Validation

LSI Latent Semantic Indexing

ML Machine Learning

MI Mutual Information

NB Naive Bayes

PCA Principle Component Analysis

PSO Particle Swarm Optimization

SVM Support Vector Machine

TC Text Classification or Text Categorization

TF Term Frequency

TS Term Strength

TN True Negatives

TP True Positives

VSM Vector Space Model

WSD Word Sense Disambiguation

Chapter 1

Introduction

1.1 Context

With rapid growth in the availability of text documents in electronic form, automatic text analysis is becoming increasingly important. One task of interest in this area is text categorization (TC), which is an important technique for organizing and understanding these data. Text categorization is the task of classifying or labelling (largely unstructured) natural language documents into one or more pre-defined categories (such as science or sport) based on their content. There are many applications for TC, largely in the context of searching and/or browsing large collections of documents [35, 40]. The ability to classify documents supports an increasing number of applications, including more informative search engine interfaces, and replacing very time consuming human effort in the manual organization of large collections of text documents. Typically, algorithms from machine learning are used to automatically classify text documents based on their content.

The basis of text/document processing is to transform a document into a term-frequency vector [125], but this immediately brings the issue of what terms, and how many terms to use to represent a document. This general question of feature selection, (FS) has a great impact in data mining in general and text mining in particular. FS has been an active research area since the 1970s [35]. In text classification in particular, feature selection aims to improve the classification accuracy and computational efficiency by removing irrelevant and redundant terms (features), while retaining features that contain sufficient information to assist with the classi-

fication task at hand. Typically there are many thousands of words that could be used as features in TC, so the need for robust FS methods is acute.

Broadly, there are two main approaches to feature selection: filter and wrapper. In the wrapper approach, typically a search is performed for an ideal subset of features, using the accuracy of classifiers (given those features) as a guide to evaluating an individual feature subset. In the filter approach, a subset of features is selected using a priori feature scoring metrics e.g. in the text categorization field features are ranked and selected in this way using metrics such as document frequency, information gain, mutual information and so forth [35, 40]. Generally, the wrapper approach is beneficial since it considers how well a group of features work together, and thus can implicitly detect and exploit nonlinear interactions among large subsets of features; however wrapper approaches are relatively slow. Meanwhile, filter approaches always have the danger of missing such interactions between two or more features, and may often discard features that may be highly relevant to the classification task.

In comparison with the English language, limited work in the text categorization field has been done for Arabic language. In particular, our aim is to improve the accuracy of TC for Arabic language texts, for applications such as automated labelling of news articles, and automated labelling or filtering of search results. In this work, we choose a wrapper approach, since we are mostly interested in developing accurate classifiers (e.g. to support a tool that post-processes the results from an Arabic search engine), and in that context it is not critical that the time spent developing the tool be particularly fast.

We note some basic differences between Arabic and English. Arabic has 28 letters and is written from right to left. In contrast with English, Arabic has a complex morphology that makes developing automatic processing systems for it a highly challenging task [84]. The basic nature of the language, in the context of text classification, is similar to English in that we can hope to rely on the frequency distributions of content terms to underpin the development of automatic text categorization. However, the large degree of inflections, word gender, and pluralities (Arabic has forms for singular, dual, and plural), means the pre-processing (e.g. stemming) stage is more complex than in the English case.

In general, most of the work done in the area of Arabic TC uses filter approaches such as chi-square and information gain for feature selection. Several studies have been reported, however this area is at an early stage. For example, although publicly available datasets exist, it is rare for any such dataset to be used in more than one work, so it is not yet possible to draw clear conclusions. The review of the work in the area of Arabic TC has shown that selection of good feature subsets (i.e. subsets that lead to good accuracy in text categorization of Arabic documents) could be well-served by investigating a wrapper feature selection approach.

In the area of Arabic text classification, [99] proposed a wrapper approach in which feature selection was performed with Binary PSO, in conjunction with Radial Basis Function (RBF) Networks. The proposed technique has been found to achieve well. Since this thesis concentrates on the feature selection issue, and state of the art machine learning, we investigated whether combinations of Binary PSO and machine learning techniques, especially K Nearest Neighbour (KNN) and Support Vector Machine (SVM) classifiers, could have a useful role in this task.

The explorations of this thesis included standard term based TC (Bag of Words), and we also investigated phrase based and combined term and phrase based approaches. We were also interested in understanding what lies behind the different performance of different approaches. This has very rarely been attempted in text classification research. Hence we also analyzed the specific sets of terms and/or phrases that were selected as features across many experiments.

1.2 Thesis Contributions

The main contributions of this thesis are as follows:

- Demonstrate successful document classification in the context of Arabic documents. We offer the used datasets to enable other researchers to compare directly with obtained results (although previous work has demonstrated text classification in Arabic, the datasets used and the experimental setup have not been revealed). Our datasets are available with full annotation (i.e. so

that other researchers can use precisely the same training and test splits) here: [112, 143].

- Demonstrate a combination of Binary Particle Swarm Optimization and K nearest neighbour that performs well in selecting good sets of features for the Arabic TC task (Section 4.3). In comparison with previous work shown in (Section 2.18.2), BPSO/KNN based FS technique for Arabic TC has shown better classification accuracy than many other published methods for particular datasets.

- We also demonstrate that using a combination of Binary Particle Swarm Optimization hybridized with Support Vector Machine for feature selection in Arabic document categorization task leads to better or similar classification accuracy in contrast with BPSO/KNN approach (Section 5.2) (Section 5.3).

- On the basis of selected features, we contribute comparative results that demonstrate the relative performance of a range of classifier methods (e.g. SVM, C4.5, and Naive Bayes) to classify a hold test set.

- We investigate the use of statistical phrases instead of single words for Arabic text representation. Although phrases have a richer meaning than single words, for Arabic TC, results show that using phrases alone for representing text documents decreased the classification accuracy of TC classifiers while using a combination of BOW and phrase based document representation leads to improved performance on certain datasets (Section 6.2) (Section 6.4).

- We also present results that show the effect of using Arabic light stemming and normalizing some Arabic letters in the pre-processing of Arabic documents on the classification accuracy of the used classifiers for Arabic TC using our FS techniques (Section 4.4) (Section 5.4).

1.3 Theoretical Aspects

The work described in this thesis is about using sophisticated optimization algorithm and sophisticated machine learning algorithms to solve a complex machine learning problem. BPSO with a machine learning algorithm is proposed as a feature selection approach for Arabic text classification problem. Binary PSO algorithm is

used to explore the feature space, and generate sets of features (candidate solutions to the given optimization problem) to find an optimal set of features (a set of Arabic terms to be used for text representation in Arabic TC task) while a machine learning algorithm (e.g. SVM) is used to evaluate the goodness of candidate solutions.

There is a body of literature of theoretical work on the topics that appear in this thesis. This includes the theory of convergence of PSO algorithms, and about how to set the parameters for PSO algorithms [56,59,144, 145]. There is also a theoretical study of generalisation in machine learning. For example, support vector machines are the result of a theoretical approach to develop an algorithm that has excellent generalisation properties [42, 45].

In this thesis, we benefit from both the theoretical and empirical advances of others, by using appropriate standard settings for a range of parameters in the optimisation algorithms, and also by making much use of SVMs, which provide the most theoretically well-grounded optimisation algorithm. However, in common with the great majority of work that attempts to solve real-world problems, the complexity of the real-world problems effectively prevents us from making significant advances in theoretical aspects in this thesis. Instead, the accumulated results of our experiments provide additional insights that can be used in future theoretical work.

However, in section (5.3), we do provide extra analyses of our results which are designed to support future theoretical work, specifically in the area of trying to understand which type of BPSO and machine-learning combination will be most appropriate for a given document classification task. Our findings, as later described, are the basis of future work that might be able to predict suitable algorithm configurations from careful statistical analyses of the dataset using text-based metrics.

1.4 Thesis structure

The thesis is organized as follows: **Chapter 2** presents a background needed to develop an efficient Arabic TC system, such as the concept of text mining in general and text categorization in particular, and techniques used to transform text docu-

ments into a form that is suitable for automatic processing. In addition, it describes the well-known machine learning classifiers for text classification. It also discusses the types of feature selection techniques and their importance in data mining. Furthermore, it introduces the Particle Swarm Optimization algorithm, its variants, and its successful applications. This chapter ends with reviewing of the related work in the Arabic TC area.

Chapter 3 describes the three Arabic datasets used in this thesis with their precise partitions into (training/test) portions. It also shows the steps of converting Arabic documents into a vector of attributes such as extracting distinct features and building the TF.IDF attribute vector.

Chapter 4 explains in detail the proposed feature selection method BPSO/KNN. It describes the pseudo code of the binary particle swarm optimization algorithm and k nearest neighbour algorithm. It shows how the suitable parameters of BPSO/KNN were set experimentally. It also reports the results of evaluating the sets of features selected by the BPSO/KNN technique on holdout test sets with SVM, Nave Bayes and J48 classifiers. It also revealed the results of comparing BPSO/KNN in two cases, without and with, including normalization and light stemming to the Arabic documents pre-processing stage.

Chapter 5 presents the second proposed feature selection method which is BPSO/SVM. It shows the results of evaluating the subsets of features selected by this method on holdout test sets in conjunction with SVM, Nave Bayes and J48 classifiers. It also compares the two proposed FS methods by analysing the sets of features selected by each FS method.

Chapter 6 is concerned with investigating the usefulness of using phrases instead of single words for representing text documents in Arabic TC. In terms of classification accuracy, this chapter also presents the results of BPSO/KNN and BPSO/SVM with a combination of single words and phrases for text representation.

Chapter 7 concludes the work of this thesis. It presents a summary of each chapter and the main findings. It also suggests some ideas for future work.

1.5 Thesis publications

- Hamouda K. Chantar, David W. Corne, "Feature subset selection for Arabic document categorization using BPSO-KNN", in proceedings of the third World Congress on Nature and Biologically Inspired Computing, Spain, 546-551, 19-21 Oct. 2011.

- Hamouda K. Chantar, David W. Corne, "Arabic text categorization via binary particle swarm optimization and support vector machines", in proceeding of the 5th Int'l Conf. on Bio-Inspired Optimization Methods and their Applications, Slovenia, 301-310, 24-25 May 2012 (Best Paper Prize).

Chapter 2

Background

This chapter presents the key concepts related to this thesis including text mining in general and text classification and its applications in particular. It also illustrates the main parts of text classification such as text pre-processing, feature selection and the most common machine learning algorithms used for text classification. It also introduces particle Swarm optimization algorithm, its variants and successful applications to solve optimization problems related to a wide range of areas. This chapter also covers the related work in Arabic text classification field.

2.1 Text mining

Nowadays, text is the most common and convenient way for information exchange. The importance of this way has led many researchers to find out suitable methods to analyze natural language texts to extract useful information. Text mining can be defined as the process of detecting meaningful and interesting linguistic patterns from natural language texts [1, 24].

In general, data mining is an automatic process of discovering useful and informative patterns in a large amount of data. In comparison with data stored in structured format (databases), information stored in text files is unstructured and difficult to deal with. To deal with such data, a pre-processing is required to transform textual data into a suitable format for automatic processing [1, 2].

In data mining, information or potentially useful patterns are usually hidden and unknown, so automatic techniques are required in order to facilitate the extraction

of these data. In text mining, the information is clear in the texts but the problem is that this information is not represented in a way suitable for processing by a computer. The Text mining field aims to represent data stored in texts in a form suitable for automatic processing [1, 2]. Text mining can be defined as applying algorithms and methods from machine learning and statistics to natural language texts to extract nontrivial information for further use [1]. Research in the text mining area involves dealing with problems such as text representation, information retrieval, text summarization, document clustering and text classification. In all these problems, data mining techniques and statistics are applied to process text data [1, 2].

Text representation is concerned with the problem of how to represent text data in appropriate format for automatic processing. In general, documents can be represented in two ways, as a bag of words where the context and the word order are neglected and the other one is to find common phrases in text and deal with them as single terms [2].

In information retrieval, the information needed to be retrieved is represented as query and the task of the information retrieval systems is to find and return documents that contain the most relevant information to the given query. In order to achieve this purpose, text mining techniques are used to analyse text data and make a comparison between the extracted information and the given queries to find out documents that include answers [1, 2].

The idea of text summarization is an automatic detection of the most important phrases in a given text document and to create a condensed version of the input text for human use [2]. Text summarization can be done for a single document or a document collection (multi-document summarization). Most approaches in this area focus on extracting informative sentences from texts and building summaries based on the extracted information. Recently, many approaches have been tried to create summaries based on semantic information extracted from given text documents [1, 2].

Text classification is the assignment of text documents into one or more pre-

defined categories based on their content [2, 5]. It is a supervised learning problem where the categories are known in advance [2]. For the text classification problem, many machine learning techniques including decision trees, K-nearest neighbour, SVM support vector machines and Naive Bayes algorithm have been used to build text classification models.

Document clustering is a machine learning technique that is used to identify the similarity between text documents based on their content. Unlike text classification, document clustering is an unsupervised method in which there are no pre-defined categories. The idea of document clustering is to create links between similar documents in a document collection to allow them to be retrieved together [1, 2, 3].

2.2 Text Classification

Due to the rapid growth of natural language text documents available in electronic form, automatic text classification becomes an important technology that is used to handle the task of organizing this data. Text classification or topic spotting is the task of assigning natural language texts into one or more pre-defined categories based on their content [2, 5]. It can be considered as a natural language problem that aims to decrease the need of manually organizing the large collections of text documents. Before using the machine learning approach for text classification problem, the most common approach for text classification task was (KE) knowledge engineering. KE is based on manually defining a set of rules that encodes expert knowledge to specify how to classify text documents based on given categories [5]. Since the early 90s, machine learning techniques have become the most common approach for the text classification problem. Machine learning for text classification problem is defined as a general inductive process that builds a text classifier by learning the features of the text categories from a set of pre-classified text documents. In contrast with a knowledge engineering approach, machine learning classifier for Text Classification task is built automatically and does not need manual definition by domain experts [5]. Figure 2.1 illustrates the general steps of the TC process.

The main three stages of text classification are text pre-processing, features

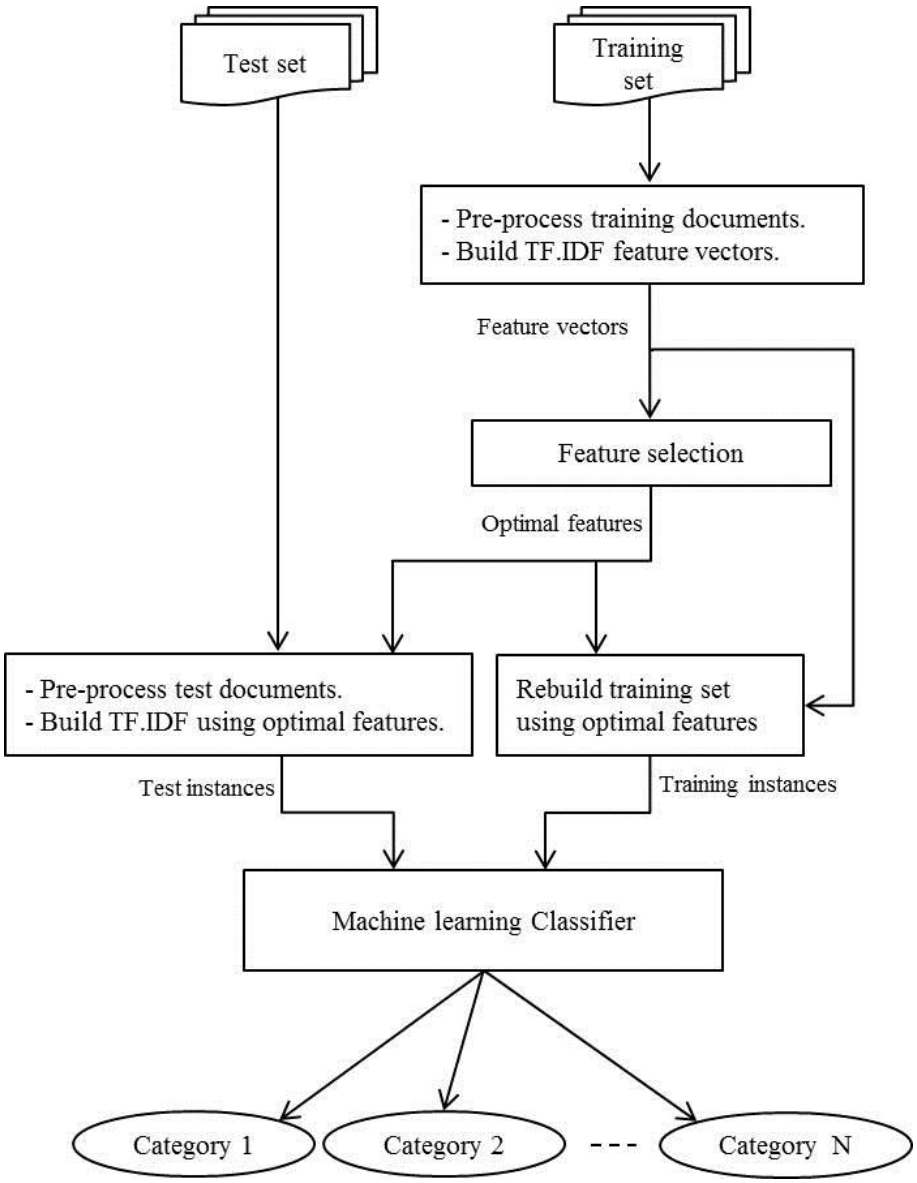

Figure 2.1: Text Classification process

selection and then building the classification model on training data to be ready for evaluation on test data. In the pre-processing, a tokenization process is performed in order to remove non-informative words such as punctuation marks and digits. Also, words that occur frequently and do not bear information (e.g. stop words and rare words) that helps in discrimination between text categories are often discarded. A set of the most informative words is kept and then used to represent text document as vectors of features. The second step is feature selection. Particularly in text classification, the goal of feature selection is to improve the classification accuracy and computational efficiency by discarding irrelevant and noisy terms (features) that do not have enough information to assist with text classification. The last stage is to build a classification model on training data using the best subset of features

selected by the feature selection and then evaluating its performance on a separate test data.

2.3 Types of Text Classification

- ***Flat and hierarchal Text Classification:*** Generally, text classification can be divided into two types, flat classification and hierarchal classification. In the flat classification, the sub-titles are not considered. When the number of documents in the category is very large, the search through such category becomes difficult and the classification accuracy of the text classification system which uses this data will be affected. This problem leads to using what is called the hieratical classification where the relationship between documents can be used by dividing each main class in the flat classification into sub-categories e.g. the Science category can be split into sub-categories such as Computer, Chemistry, Physics, etc [3].

- ***Single-label and multi-label Text Classification:*** Single-label in text classification is the process of assigning a document d_j in a given dataset into only one of pre-defined categories while in multi-label task, a document d_j can be assigned to zero, one or more than one category [3, 6]. Choosing between single-label and multi-label depends on whether the application is a single-label or multi-label text classification problem [6].

- ***Soft and hard Text Classification:*** Deciding whether a specific document d_j belongs to a class C_i or not, is called a hard classification decision or a binary decision. Another type of decision is called soft decision. In this case, a numeric score in a specific range is used to measure the degree of the classifiers confidence that a document d_j belongs to a class C_i [6].

2.4 Applications of Text Classification

The text Classification approach has been found highly beneficial for many applications. Among these applications are:

- ***Web page Classification:*** The number of web-pages available on the World Wide Web is growing rapidly. The task of searching for specific information

without organizing web pages is difficult [22]. It is obvious that performing web page classification manually is unfeasible. So, the need for efficient automatic web-page classification systems is increased. For automatic web classification, TC techniques have been utilized for classifying web pages under hierarchal categories to facilitate the web navigation [5]. When web pages are organized in this way, it is easier and faster for a web search engine to start navigating the hierarchy of categories and then limiting its search in the category that contains the required information [5].

- ***Word Sense Disambiguation:*** Word sense disambiguation (WSD) is the process of finding the correct meaning (sense) of a word in a giving context [5, 17]. WSD is essential for many tasks related to natural language processing such as query based information retrieval, machine translation, and information extraction [5, 17]. Text classification techniques have been applied successfully to the WSD problem where ML classifiers can be constructed from semantically annotated corpora, and then used to select which word sense is appropriate for a given context [17, 18].

- ***Text filtering:*** Text filtering is the process of classifying incoming text documents into separate categories based on the users interest. Text filtering can be seen as a single label text classification task [5]. An example of text filtering is the e-mail spam filtering system. Anti-spam email filter based on the machine learning algorithm can automatically learn from previously received emails on how to distinguish between spam and non-spam messages [20].

2.5 Text pre-processing

Document pre-processing is an essential step in text mining. The aim of the pre-processing stage is to transform text documents into a suitable format for automatic processing [7,24, 25]. The most common steps for text pre-processing are:

- ***Tokenization:*** Documents are tokenized by removing punctuation marks, digits and numbers. The remaining character strings are considered as terms or tokens [1, 24].

- ***Stop words removal:*** Words such as prepositions and articles that occur

frequently and do not help in discrimination between classes are called stop words or function words [7, 24]. These words are usually removed using a stop words list. Stop words are language dependent e. g. English stop words include the, other, or, in, there, etc [7, 21].

- ***Stemming:*** Stemming is the process of reducing words to their stem or root form where morphological information is used to match different variants of words [7, 21]. For example, the words *read*, *reading* and *reader* can be reduced to their root *read*. In general, stemming is an important step in natural language processing, text mining and information retrieval related applications [27,28, 29]. In the text classification problem, stemming could be used as a dimension reduction technique to deal with high dimension feature space problem to enhance the accuracy of text classification systems. There is a kind of stemmer called a light stemmer. This stemmer does not affect the semantics of words. It removes some prefixes and suffixes. For the Arabic language, light stemmer strips off some prefixes and suffixes of Arabic words [27, 28].

  For Arabic TC task, [26,27, 28] used light stemming as a feature selection method and found that light stemming works well as a feature selection technique for Arabic TC.

  In this work, light stemmer (light10) was used. This stemmer was developed by [29]. It has been widely used in applications related to Arabic text processing and Arabic information retrieval applications, and has been included in Apache Lucene open source project [30]. Apache Lucene is a full-featured text search engine library written in Java. Java libraries for Arabic normalization and stemming in Apache Lucene are available at [31, 32]. Normalization and stemming operations were added to the text pre-processing steps. Normalization is performed according to the following steps [29]:

  - Replace Alef "أ ، إ" and " آ " with "ا".
  - Replace Yeh "ي" with "ى".
  - Replace "ة " with "ه".

– Remove Arabic diacritics and stretching character (Tatweel).

Arabic Light stemmer (light10) does not affect the meaning of words. It only removes the conjunction "Wa" " "و" , some prefixes like:

(ف ، ك ، ل ، ب ، ال ، وال ، بال ، كال ، فال ، لل)

and some suffixes such as:

(ها ، ون ، وا ، ين ، ان ، يه ، ية ، ا ، ة ، ه) [31].

- ***Rare words removal:*** Usually, the number of distinct terms after removing stop-words and stemming is still large, and most of the distinct terms appear rarely [7, 21]. Low frequency words are not helpful in discrimination between text documents because it is hard for text classification models to learn such terms [7]. It is found beneficial to remove words with frequency less than a pre-defined threshold e.g. two or three times [7].

  [8, 9] studied the effect of rare words removal, stop words removal and stemming on the text classification task. They found that stop words removal is an important step. It eliminates the terms that could affect the classification accuracy significantly. They also showed that stemming may decrease the classification accuracy slightly. Reference [8] also reported that rare words removal has almost no impact on the classification accuracy. However, both conclude that stop words removal and stemming may lead to a great reduction of the feature space dimension.

2.6 Text representation

After extracting the distinct words for text documents in the pre-processing stage, the next step is to represent the selected features in a suitable format for automatic processing. The most common for document representation in the TC task is called BOW bag of words. Also, there is another approach based on phrases that can also be used for representing text documents [10, 12].

2.6.1 Bag of words representation

Bag of words or also known as Vector space Model is the most popular and simplest way of text representation [8,4]. Term or feature in BOW representation is a single word. Each text document is represented as a vector of weights $d_j =< w_{1j}, w_{2j}, w_{3j}, ..., w_{Nj} >$ where N denotes the number of distinct features (terms) in the document collection [4].

2.6.2 Phrase based representation

Most text representation used in text mining, information retrieval and related applications is the bag of words approach (BOW). A number of researchers have tried to use phrases instead of or as well as single words as features to represent text documents [10, 11]. In general, two different types of phrases have been proposed and investigated:

- ***Syntactic phrase:*** Syntactical phrases such as noun phrases and verb phrases can be obtained from text documents based on syntactical rules [10, 12].

- ***Statistical phrase:*** After eliminating stop-words, statistical phrase is a sequence of n words occurring consequently in the context [10, 11].

  The advantage of using phrases for text representation is that phrases have larger meanings than single words [11, 12]. In terms of classification accuracy, most of the work done using different forms of phrases for text representation did not show better or encouraging results in comparison with single word representation [10,11, 16].

  Some authors have investigated the effect of using phrases to represent text documents for text classification task. For instance, in [10], after stop words removal, a statistical phrase was defined as stemmed and alphabetically ordered a sequence of n words. Statistical phrases of different lengths of n-grams were extracted. Unigrams and bigrams were used to represent text documents. Also, different filter approaches were used as features selection techniques. Moreover, the Racchio algorithm was used as a text classification model. Experimental results on the Reuters-21578 benchmark show that using statistically extracted unigrams and bigrams as features did not yield better

results than using single word representation.

In [11], based on the notation of aboutness used in the information retrieval field, a dependency model was used to capture dependency triples to be used as features in text classification. Dependency triples represent syntactic phrases and dependency graphs represent the structure of sentences. SVM and Winnow were used as text classifiers. The results show that combining such triples with bag of words representation leads to significant improvement of document classification accuracy.

The authors in [12] investigated the usefulness of using multi words for text representation in text classification problem. Based on syntactic rules, an algorithm was used to extract the repetition patterns of two sentences and then, regular expression was employed to extract noun phrases to be used for representing text documents. In addition, two different strategies based on extracted multi words were used. Also, information gain was applied as a feature selection method and SVM was used as a text classifier. A series of experiments was done to classify the Reuters-21578 documents using the proposed method. In comparison with single word representation, the results show that using multi words for text representation did not yield better classification accuracy.

Also, [14] used BOW and phrases for text representation for classifying text documents. In comparison with using BOW representation, using both BOW and phrases for text representation did not show significant improvement in terms of classification accuracy.

In [15], WordStat tools and SAS Enterprise Miner were used to extract single key words and phrases from two datasets formed randomly from 20Newsgroup and OHSUMED datasets. Decision trees, Neural Networks and Memory Based Reasoning classifiers were used as classification models. Also, three text representations include keywords, phrases and both keywords and phrases were tested. Experimental results show that using phrases alone or keywords and

phrases together for text representation improves the classification accuracy.

For Arabic language text classification, [85] used unigram and bigram together in term indexing for classifying Arabic documents and compared the obtained results with unigram indexing only (BOW). KNN was used as a text classifier. The average accuracy using single term indexing (BOW) is 0.668 while the average accuracy using both unigram and bigram is 0.735. The results show that combining unigram and bigram for term indexing is better than using BOW alone.

2.7 Term Weighting

In the text classification problem, terms that appear in documents are represented to machine learning classifiers as real-numbered vectors of weights. The most commonly used methods are Term Frequency (TF) and Term Frequency Inverse Document Frequency TF.IDF [4, 8].

- ***Boolean weighting:*** Boolean weighting is the simplest way for weighting the terms. If a term t_i occurs in a document d_j at least once then its weight is 1 otherwise 0 [4, 8].

- ***Term Frequency TF:*** In term frequency weighting scheme, the weight of a term t_i in the feature space is the number of times that the term appears in a specific document d_j [4, 8].

- ***Term Frequency Inverse Document Frequency TF.IDF:*** TFIDF can be considered as a statistical weighting scheme. It is a common method used in the text classification problems and related applications such as information retrieval. It is a simple and effective method for weighting the terms in text documents for classification purposes [8,19, 24].

  Document frequency of a term DF (t_i) is the number of text documents in the corpus in which t_i occurs at least once and the inverse document frequency IDF of the term t_i can be calculated as follows [8]:

$$IDF(t_i) = \log \frac{D}{DF(t_i)} \tag{2.1}$$

Where, D is the total number of documents in the dataset. The weight of term t_i in a document d_j using $TF.IDF$ is defined as:

$$TF.IDF(t_i, d_j) = TF(t_i, d_j) * IDF(t_i) \tag{2.2}$$

2.8 Dimensionality Reduction

The term "curse of dimensionality" was proposed by Richard E. Bellman in 1961. It refers to the problem of data with high dimension feature space [133]. Since many pattern recognition and classification techniques are not able to cope with problems with high dimension of features. Researchers have developed many dimension reduction techniques that aim to reduce the dimension of feature space by eliminating noisy, irrelevant, and non-informative data while retaining relevant and informative ones [40, 133]. Generally, dimension reduction approaches are classified into two classes: feature extraction and feature selection.

In feature extraction approach, the original set of feature is transformed into a lower feature space using liner transformation techniques such as principle component analysis (PCA), liner discriminant analysis (LDA) and latent semantic indexing (LSI) [132, 133]. PCA uses subspace projection approach to find basis vectors that best minimize the mean square error [133, 135, 136]. The basic idea of LDA is to search for a linear function of data vectors that well differentiate between classes [133, 137]. Latent semantic indexing (LSI) was originally developed by [138] as term indexing approach for information retrieval. Using Singular Value Decomposition (SVD), the original high feature space can be transformed into a lower semantic feature space. In this sense, truncated SVD is used to capture highly associative patterns (word-text associations) in the data, and discards noisy patterns [138, 139]. LSI has been successfully applied to TC problem [139, 140, 141].

In contrast with feature extraction, without transforming the feature space, feature subset selection (FSS) or feature selection (FS) is the process of selecting a subset of the original set of features [40, 133]. In this work, FS has been adopted for Arabic text classification problem. We will discuss FS in more details separately.

2.9 Feature Selection

Feature selection has a great impact in data mining in general and text mining in particular [34]. FS is a basic problem in pattern recognition and classification as most existing learning algorithms are not designed to deal with high dimension of feature space. It has been an active research area since the 1970s [35]. In the text classification problem, feature selection aims to improve the classification accuracy and computational efficiency of pattern recognition and classification techniques (e.g. Text classification models) by removing irrelevant and redundant terms (features) from the corpus. It is also used to select features that contain sufficient information about the text dataset.

FS has two wider approaches: wrapper and filter. In the wrapper approach, a subset of the features is selected depending on the accuracy of the classifiers while in the filter approach, a subset of features is selected or filtered using feature scoring metric [33,34,38, 40].

Figure 2.2 shows the relation between the dimension of data and the accuracy of the classifier [133]. Practically, after exceeding a specific number of features, the performance of the classification or recognition model will decrease gradually.

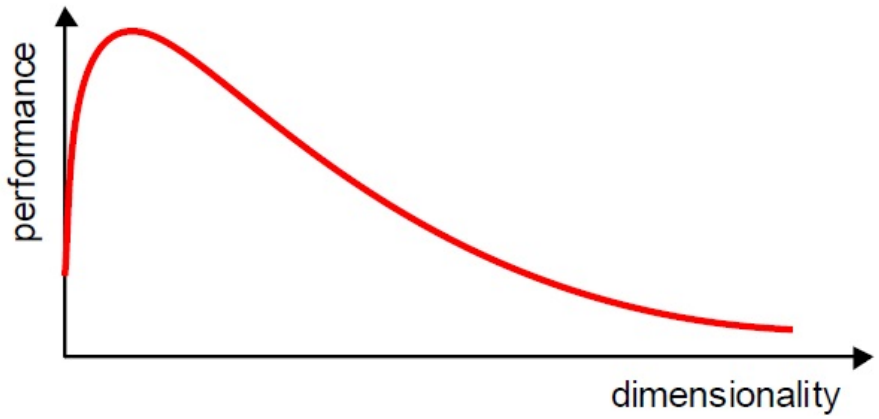

Figure 2.2: Relation between the dimension of feature space and the performance of the classification model [133]

Feature selection has several objectives in the area of pattern recognition, data mining and machine learning. Some of these objectives include [40, 133]:

- To help in building fast and accurate pattern recognition and classification

models by discarding noisy and redundant features.

- Remove unwanted features leads to lessening the run-time and reduce required storage media.

- In case of feature extraction, transformation and projection may lead to loss of properties of features. In contrast, Feature selection does not affect the original features set, using feature selection makes the process of extracting interpretable and meaningful rules from the data possible.

- Enhance the generalization ability of pattern recognition and classification models.

2.9.1 Feature selection process

Feature selection process requires two main steps: search strategy and objective function [133]:

- *Search strategy*

  The aim of this step is to generate subsets of features for evaluation by an objective (evaluation) function. The behaviour of the search process usually depends on the initial start point of the search. There are three ways to start the search through feature space: forward selection, backward elimination, and random selection. In forward selection, the search begins with a set with zero features and then, features are added gradually based on their goodness. In backward elimination, the search starts with a set that contains all features then, unwanted features are eliminated. The search could also start by selecting subsets of features randomly. In the random search, the best subset of features is selected based on an evaluation (fitness) function [132, 133, 134]. In common, search strategies in feature selection field are broadly categorised into three main classes: random, complete, and sequential. In sequential search, features are eliminated or added in sequential way. Examples of sequential search approaches are forward selection, backward elimination and, bidirectional search [132,133, 134]. In the complete search strategy, all possible subsets of features are generated and evaluated to find the best one (e.g. beam search). The random search strategy arbitrarily generates subsets of features and evaluates them to determine the best subset among them.

Examples of random search techniques are random-start-hill-climbing, genetic algorithms, Ant Colony Optimisation algorithm, and PSO algorithm [132,133, 134].

- *Objective function*

  In FS process, an Objective function is used to measure the goodness of subsets of features generated by a search algorithm. There are two types of objective functions: filters and wrappers. In filter approach, statistical methods such as chi-square, information gain are used to select top ranked features from a given dataset and remove low ranked ones. In wrapper approach, classification models are applied to measure the quality of candidate subset of features. In this way, features with best classification accuracy (measured by the used classifier) are chosen for data representation [133, 134].

2.9.2 Filter approach

In the filter approach, a subset of features is selected using feature scoring metrics. Only features with high scores are retained while features with low scores are considered as irrelevant features, so they are discarded [40]. The filter approach is widely used in the text classification field where features are chosen by scoring matrices like document frequency DF, information gain IG, mutual information, chi square CHI, correlation coefficient and term strength TS [34,36, 37]. Figure 2.3 describes the filter approach. It can be seen that the feature subset selection process is performed first; then the selected subset of features is used to evaluate the classification model. In this case, feature subset selection is done separately and it does not have an interaction with classification models [39, 40].

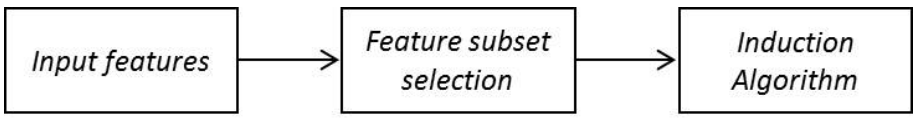

Figure 2.3: The filter approach model [39]

Filter approaches are computationally fast, simple and have the ability to deal with high dimension feature space [40]. The drawback of the filter approach is that it does not assess the effect of selected features on the performance of the classification

model [39, 40].

- ***Document Frequency (DF):*** *DF* threshold is a simple with low cost in computation feature selection method. The idea is that low frequency words are not helpful or irrelevant for class prediction. For each distinct (word) feature in the document frequency method, its *DF* is the number of documents in which the feature happens with a threshold. In other words, the value of a specific feature is the number of documents containing that feature. Then, removing from the feature space all terms which their document frequencies are less than a pre-defined threshold. *DF* is defined as [35, 36]:

$$DF = \sum_{i=1}^{m}(A_i) \tag{2.3}$$

- ***Information Gain (IG):*** Information gain method is frequently employed. Usually, the information gain for each term is computed and the terms with *IG* less than a pre-determined threshold are removed [33]. In the *IG* method, the goodness measure of a term for class prediction can be estimated by the presence or absence of that term in a document. Let C be a set of classes in a training dataset, the information gain of a term (t) can be estimated as [33]:

$$\begin{aligned} IG(t) = &-\sum_{i=1}^{m} P(C_i)\log P(C_i) \\ &+ p(t)\sum_{i=1}^{m} P(C_i|t)\log P(C_i|t) + p(\neg t)\sum_{i=1}^{m} P(C_i|\neg t)\log P(C_i|\neg t) \end{aligned} \tag{2.4}$$

  Where $P(c_i)$ is the probability documents in the dataset belong to a class (c_i); $P(t)$ and $P(\neg t)$ are the probability that a term (t) is in the documents of the dataset or not; and $P(c_i|\neg t)$ is the probability that documents in the class (c_i) contain a term (t). m is the number of features [33].

- ***Mutual Information (MI):*** Mutual information is a method used in statistical language modelling of word association. It determines the mutual dependency between a term t and a class c [33, 36]. Using the two way contingency table, suppose we have a term t and a document class c, A denotes the number of times that t appears in c, B denotes the number of times that

t happens in all classes except c, C denotes the number of times c happens without t, and N is the total number of text documents, then MI is determined as [36]:

$$MI(t,c) = \log \frac{P(t,c)}{2p(t)p(c)} \tag{2.5}$$

$$MI(t,c) = \log \frac{A \times N}{(A+C) \times (A+B)} \tag{2.6}$$

- ***Chi square Statistic (CHI):*** Chi square method is used to determine the degree of dependency between a term t and a document class c. Base on the two way contingency table, suppose we have a term t and a document class c, A denotes the number of times that t appears in c, B denotes the number of times that t happens in all categories except c. C refers to the number of times c happens without t, D represents the number of times that c and t do not exist, and N is the number of all documents, then *Chi-square* is determined as [33, 36]:

$$Chi(t,c) = \frac{N \times (AD - CB)^2}{(A+C)(B+D)(A+B)(C+D)} \tag{2.7}$$

$N = A + B + C + D$.

If $\chi^2(t,c)$ is equal zero that means the term t and the category c are independent. One of the ways to determine the term goodness of t by calculating the chi square of that term with each of all given categories and then picking the maximum score to be the chi square value of t [36].

- ***Correlation Coefficient (CC):*** Correlation coefficient can be seen as one-sided chi square matrix. It is defined as [37]:

$$CC(t,c) = \frac{(AD - BC)\sqrt{(A+B+C+D)}}{\sqrt{(A+B)(C+D)(B+D)(A+C)}} \tag{2.8}$$

- ***Term Strength (TS):*** Term strength is used for feature reduction in the text classification task. TS estimates the importance of a term based on that term appearing in pair related documents. It calculates the probability that a term occurs in a pair of documents. Let d_1 and d_2 be related documents, and t is a term then the TS of the term t can be estimated as follows [36]:

$$TS(t) = P(t \in d_1 | t \in d_2) \tag{2.9}$$

2.9.3 Wrapper approach

In the wrapper approach, a search is performed for an ideal subset of features, using the accuracy of the classifiers (given those features) as a guide to evaluating an individual subset of features [38]. Figure 2.4 describes the wrapper approach. The feature subset selection algorithm works as a wrapper around the induction algorithm [38, 40]. It uses a heuristic search technique (e.g. particle swarm optimization algorithm) to generate a different subset of features; and then an induction algorithm is used to evaluate each subset of features separately [39, 40]. The subset of features with highest evolution is selected as the best subset of features and then is used to build the classification model. Afterwards, a separate dataset that was not engaged in the selection of the best subset of features is used to evaluate the classification model [38].

Generally, the wrapper approach is beneficial since it considers how well a set of features work together and thus can implicitly detect and exploit nonlinear interaction between large sets of features; however wrapper approaches are relatively slow. Also, in comparison with filter approaches, wrapper techniques are computationally intensive and have a higher danger of over fitting [40].

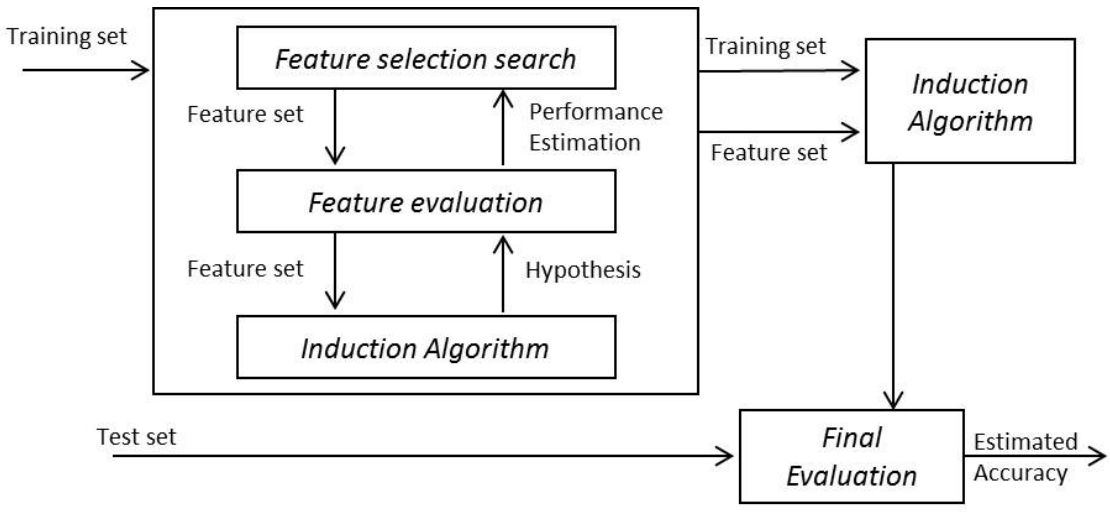

Figure 2.4: Wrapper Approach [38]

2.9.4 Comparison between filter and wrapper feature selection approaches

Table 2.1 summarizes the main advantages and disadvantages of filter and wrapper feature selection approaches [40, 133].

	Filter	**Wrapper**
Advantages	-Fast. -Better generality than wrappers. -Does not interact with the classifier. -Less computational complexity than wrappers. -Scalable to high dimension data.	-Interacts with classifiers. -Does not ignore features dependency.
Disadvantages	-Omits features dependency. -Omits interaction with classifiers.	-Costly in computation. -Slow. - Risk of getting trapped in local optima. -Suffers from danger of overfitting.

Table 2.1: Main advantages and disadvantages of filter and wrapper approaches

In this work, we choose a wrapper approach since we are mostly interested in developing accurate classifier (e.g. to support a tool that post-processes the results from an Arabic search engine), and in that context it is not critical for the processing time to be particularly fast.

2.10 Machine Learning Classifiers for Text Classification task

The most popular and successful machine learning algorithms which are frequently used for text classification include SVM, Naive Bayes, K Nearest Neighbour and Decision Trees.

2.10.1 Decision Trees Classifier

Decision Tree is a machine learning classifier that takes the form of a tree where a collection of training instances constructs the classification tree. The most common decision tree algorithm is C4.5. It is an extension of the earlier version of decision tree algorithm ID3. Leaf nodes of the tree represent the class labels or target classification. The mechanism of decision tree classifier for classifying unseen instances is to test at each node some feature values to determine the class of a given unseen instance. The test begins at the root node and goes down until a leaf node is reached where that leaf node indicates the class of unseen instance [23, 44].

Figure 2.5 is an illustrative example of a decision tree for the set of training examples shown in Table 2.2. Let $< a_1, b_2, a_3, b_4 >$ be an unseen instance. Based on the constructed decision tree, the classification of unseen instance is yes [44].

F1	F2	F3	F4	Category
a1	a2	a3	a4	Yes
a1	a2	a3	a4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
c1	c2	c3	a4	Yes
c1	c2	c3	c4	No
c1	c2	c3	c4	No
c1	c2	c3	c4	No

Table 2.2: Set of training examples [44]

The root node of the decision tree is the feature that best splits the training

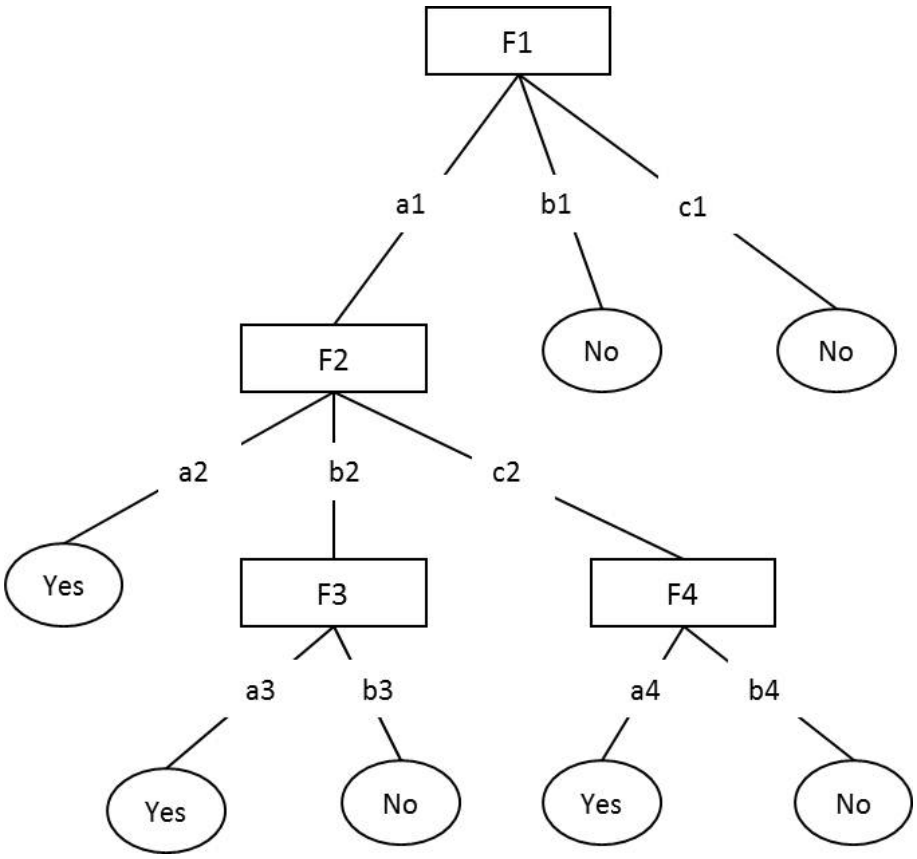

Figure 2.5: Example of Decision tree [44]

examples. Information gain IG is a good measure for picking up the best feature where the feature with highest information gain is selected to be the root node [23, 44].

Decision trees have been used for solving a wide range of practical problems such as assessing credit risk of loan applicants, classifying medical cases according to patients disease, detecting advertisements on the web and identifying spam emails [23, 48].

Decision trees suffer from the problem of overfitting. According to [23], the precise definition of overfitting is

> *"Given a hypothesis space H, a hypothesis $h \in H$ is said to overfit the training data if there exists some alternative hypothesis $h' \in H$, such that h has a smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances".*

The issue of overfitting in decision trees could be avoided by pruning the tree after

its full growth or preventing it from reaching its perfect size and fitting the training examples [23, 44].

2.10.2 K Nearest Neighbour (KNN)

KNN learning algorithm is widely used in text classification tasks due to its effectiveness and robustness [41]. It is an instance based learning method that simply stores training instances as points in N-dimensional space. When a new instance (x) is to be classified, a set of similar training instances is retrieved and used to predict the class of the new instance. KNN finds the K closest instances in the training set to x and assigns the most common class of the nearest neighbours to x. The standard Euclidian distance function is often used to measure the similarity between two instances. It is defined as the following [23]:

$$d(x, y) = \sqrt{\sum_{i=1}^{N} (a_i(x) - a_i(y))^2} \tag{2.10}$$

Where x and y are two points in N-dimensional space and a_i is the value of the i_{th} attribute. Figure 2.6 shows how *KNN* classifies an unseen instance to one of two classes according to its K nearest neighbours.

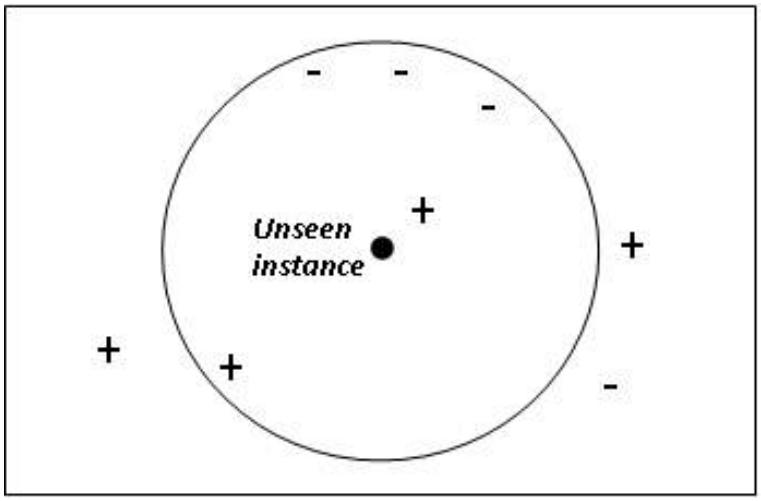

Figure 2.6: Example of classification using KNN classifier [23]

If K parameter of *KNN* is set to one then the unseen instance will be classified as a positive instance. Where K is equal to five, the classification of the unseen instance is negative. Moreover, in addition to Euclidian distance measure, different distance measures could be used with *KNN* classifier to determine the K nearest

neighbours to unseen instances. Table 2.3 describes the most common ones [44,98, 110].

$Manhattan(x_i, y_i) = \sum_{i=0}^{n} \lvert x_i - y_i \rvert$
$Cosine = \frac{x.y}{\lvert\lvert x \rvert\rvert \lvert\lvert y \rvert\rvert}$
$Jaccard = \frac{\lvert x \bigcap y \rvert}{\lvert x \bigcup y \rvert}$
$Dice = \frac{2\lvert x \bigcap y \rvert}{\lvert x \rvert + \lvert y \rvert}$

Table 2.3: Similarity measures

x and y are two vectors in N-dimension space.

2.10.3 Naive Bayes classifier (NB)

Naive Bayes classifier is a probabilistic classification model, based on Bayes theorem. It is a simple and practical classifier [23, 46]. NB lies on the assumption that the values of features are conditionally independent given target classes [23].

Consider a set of training examples where each example X is described by a set of feature values $< F_1, F_2, ...F_n >$ and target classification. Let C be a set of classes defining the target function. Given a test example t, based on the feature values, NB assigns the test example to the class with the highest probability [23].

The probability that the test example t belongs to a specific class C_j can be estimated as follows:

$$P(C_j|t) = \frac{P(t|C_j)P(C_j)}{P(t)} \tag{2.11}$$

$P(C_j|t)$ is the probability of the class C_j given a test example t. *P(t)* is equal for all categories so, it can be ignored [46].

$$P(C_j|t) = P(t|C_j)P(C_j) \tag{2.12}$$

Using the assumption from Bayes theorem that says the features are conditionally independent, the probability of class C_j can be rewritten as below [23]:

$$P(C_j|t) = P(C_j) \prod_{i=1}^{n} P(f_i|C_j) \tag{2.13}$$

N is the number of features (f_i) that form training examples. The class of the test instance t is determined by NB classifier:

$$V_{NB} = argmax_{c_j \in C} P(C_j) \prod_{i=1}^{n} P(f_i|C_j) \tag{2.14}$$

V_{NB} is the output of Naive Bayes classifier which refers to the class of test instance.

Although the features independence assumption is unrealistic, Naive Bayes has been found very effective for many practical applications such as text classification and medical diagnosis. This is due to its ability to scale with high dimension feature space [23, 46, 47].

2.10.4 Support Vector Machine (SVM)

Support Vector Machine is a universal machine learning technique. It is based on the structural risk minimization principle where input points in N-dimensional space are mapped into a higher dimensional space and then a maximal separating hyperplane is found [42]. Consider a training set of labeled instances $x_i \in R^n, i = 1...L$, belong to a set of categories $y_i \in |-1, 1|$. Figure 2.7 is an example of an optimal hyper plane for separating two classes [45]. From Figure 2.7, SVM builds the classification model on the training data using a linear separating function to classify unseen instances [44].

For linearly separable vectors, the kernel function is simple. It takes the form:

$$f(x) = W.X + b \tag{2.15}$$

W is called weight vector for optimal hyper-plane and b is known as the bias. The class of X (test instance) can be found using the following linear decision function [42, 44]:

$$y = sign(f(x)) \tag{2.16}$$

The optimal separating hyper plane is the one that has the largest margin. The

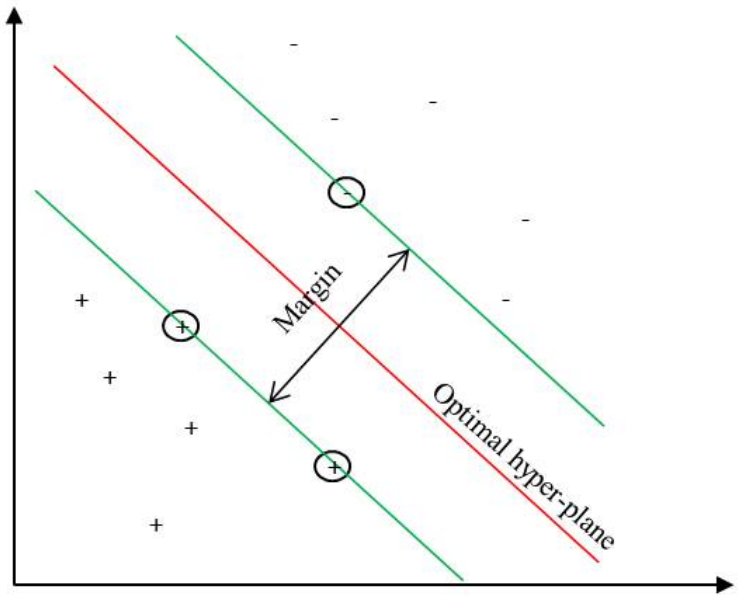

Figure 2.7: linear separation between two classes (points marked with circles are support vectors) [44]

distance between nearest vectors to the hyper plane is maximal. The distance is given by:

$$min_{x_i,y_i=+1}\frac{(w.X)+b}{|w|} - max_{x_i,y_i=-1}\frac{(w.X)+b}{|w|} = \frac{2}{|w|} \tag{2.17}$$

The hyper plane which minimizes W is considered as the optimal hyper plane [45].

$$\frac{1}{2}||w||^2 \tag{2.18}$$

Using Lagrangian formula, the maximal margin hyper plane can be rewritten as:

$$f(x) = \sum_{i=1}^{n} a_i y_i k(x, x_i) + b \tag{2.19}$$

Where, $k(x, x_i)$ is the kernel function, y_i is the class label of support vector x_i, x is a test vector, a_i is a Lagrange multiplier for each training vector (vectors for which $a_i > 0$ are called support vectors), b is a numeric parameter (scalar) and n is the number of support vectors. In the text classification problem, x_i represents the i_{th} document in the training set and y_i denotes the class of that document (e.g. sport, science, religion, etc) [42, 45].

For non- linearly separable data, different kernel functions can be used with *SVM*. The most common kernel functions are [42, 43]:

- Polynomial kernel:

$$K(x_i, x_j) = (\gamma x_i.T.x_j + r)^d, \gamma > 0 \tag{2.20}$$

- RBF kernel:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2), \gamma > 0 \tag{2.21}$$

- Sigmoid kernel:

$$K(x_i, x_j) = tan(\gamma x_i.T.x_j + r) \tag{2.22}$$

Here, γ, r and d are kernel parameters.

When the number of features is very large as in the document classification problem, the linear kernel function is the proper choice and there is no need to map the data [43].

2.10.5 Fuzzy C-means(FCM)

Clustering concerns grouping a set of given unlabelled instances (represented as points in N dimension space) $X = x_1, x_2, ..., x_N, x_i \in R^f$ into a set of clusters, such that, similar instances (e.g. text documents) are assigned to the same cluster [13]. Fuzzy C-means is one of the popular clustering techniques [13]. FCM uses a given data points to randomly generate an initial set of cluster centers. The idea of FCM is to minimize an optimization (objective function) which is used to evaluate the quality of the partitioning that splits the given data points into clusters [13, 142]. The objective function is given as follows [13]:

$$J_m(U, W) = \sum_{j=1}^{C} \sum_{i=1}^{N} (u_{ij})^m d_{ij}^2 \tag{2.23}$$

N is number of instances in X, C is number of clusters; U is the Matrix of the membership function; W is vector of the cluster center and m controls the degree of overlap between clusters. The objective function is restricted with the following constraints:

$$u_{ij} \in [0, 1], \sum_{j=1}^{C} u_{ij} = 1, 0 < \sum_{i=1}^{N} u_{ij} < N \tag{2.24}$$

u_{ij} is the member functions value and d_{ij} is the distance between x_i and w_j. Fuzzy C-Means technique has been used for document clustering. An examples of that is presented in [142].

2.11 Performance Evaluation

In data mining, evaluation of the accuracy of machine learning algorithms is an essential step. To classify given data, a set of training data is used to build the classification model. For estimating the accuracy of the obtained classifier, two common approaches, hold-out and cross-validation are used to assess the ability of the classifier to predicate the correct class or category of unseen instances.

2.11.1 Hold out

In the hold out method, the available data is arbitrarily split into two separate sets, a training set and a test set. Usually two thirds of the data are retained for training and the remaining third is used for testing. A problem may arise when one of the classes is not represented in the training portion of data. This problem is solved using what is called stratified hold out. In this case, the selected sample contains instances from all classes of the data. In other words, all classes are represented in both data sets [24, 25].

2.11.2 K-fold-Cross validation

In this method, the data is randomly split into K equal subsets or folds. Repeating K times, each subset is used for testing and the other remaining folds for training. Then, overall error is estimated by averaging the K errors. Usually, the stratified version of K fold cross validation is used to ensure that all given classes are represented in all folds [23,24, 25].

2.11.3 Leave One Out Cross Validation (LOOCV)

Leave one out cross Validation is similar to K fold cross validation. The difference is that the number of folds is equal to the number of instances in the dataset. This means that at each run, there is only one instance in the test set. The advantage of the LOOCV technique is that it avoids random sampling. All training data participate in the learning algorithm training; however, this method is computationally costly [24, 25].

2.11.4 Error rate

Error rate is the percentage of misclassified instance in a given test set. Consider a test set D consists of N instances, and r is the number of misclassified instances by a classifier. The accuracy of the classifier for correctly predicting the classes of the instances in D can be estimated as follows [23, 24]:

$$Acc = \frac{r}{N} \tag{2.25}$$

For more reliable estimation, normal distribution is used to estimate the accuracy. In case the dataset size is not small, the estimated accuracy is given as below:

$$P = z\sqrt{\frac{(Acc)(1-Acc)}{N}} \tag{2.26}$$

The accuracy is in the range:

$$Acc = Acc \pm P \tag{2.27}$$

The disadvantage of the error rate method is that it ignores the cost of wrong prediction which is important in machine learning. This problem can be avoided using *F-measure* [24].

2.11.5 F1-measure

F-measure is widely used in the information retrieval field and is calculated based on two measures, precision and recall. In this context, consider the documents in

the test set that is category A. The classifier predicts a category for each document, and these predictions will fall into four classes with respect to category A [24, 25].

- ***True Positives (TP):*** the set of documents that are in category A, and were correctly predicted to be in category A.
- ***True Negatives (TN):*** the set of documents that are not in category A, and were predicted to be in a different category than A.
- ***False Positives (FP):*** the set of documents that were predicted to be in category A, but in fact they are of a different category.
- ***False negatives (FN):*** the set of documents that were predicted not to be in category A, but are actually in category A.

Precision is the proportion of predicted category A documents that were correctly predicted.

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{2.28}$$

Recall is the proportion of actual category A documents that were correctly predicted.

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{2.29}$$

The *F-measure* is the harmonic mean of precision (p) and recall (r).

$$F\text{-}measure = 2 \cdot \frac{precision.recall}{precision + recall} = \frac{2 \cdot |TP|}{2 \cdot |TP| + |FP| + |FN|} \tag{2.30}$$

2.11.6 Confusion Matrix

This is a simple way to view the performance of classification algorithms. Consider a problem of two classes; using the confusion matrix, the actual and predicted classes of the test set instances can be displayed as in Table 2.4 [24, 25]:

The accuracy and error of the classifier are calculated as the following:

$$error = \frac{FP + FN}{TP + FP + TN + FN} \tag{2.31}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.32}$$

	Predicted	
*Actua*l	Yes	No
Yes	*TP*	*FN*
No	*FP*	*TN*

Table 2.4: Confusion Matrix of two classes

2.11.7 T-test

T-test is a statistical method that is used to measure the difference between the means of two sets. In machine learning, T-test can be applied to assess the performance of two learning techniques to see whether the difference between their accuracy means is statistically significant or not [23].

2.12 Machine Learning Software

There are several machine learning software tools and libraries available for research purposes [50, 100]. The most commonly used tool by machine learning research community is Weka ML software [50]. Weka has been chosen for the implementation in this work for the following reasons:

- The author has previous experience in using Weka and, is familiar with Java programming.
- All needed machine learning classifiers for this work are available in Weka.

Weka refers to Waikato Environment for Knowledge Analysis and is well-known machine learning software. It is developed at Waikato University in New Zealand. *Weka* is written in Java, contains a collection of machine learning algorithms as well as tools for data pre-processing and analysis. *Weka* provides tools and algorithms for data mining like data pre-processing, visualization, feature selection, classification regression and clustering [24, 49]. *Weka* is an open source and free software available at [50].

ARFF (Attribute Relation File Format) is the standard way of formatting and storing data in *Weka*. It is an ASCII text file which contains a set of instances

represented by a list of features. It also specifies the types of attributes and the classes of instances [24,49].

The general format of the *ARFF* file is as in the example shown in Figure 2.8. *ARFF* file has two main sections, relation and data.

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
```

Figure 2.8: Example of Weka ARFF file [49]

The relation section starts with the relation name and then, defines list of attributes. The last part of the relation section defines the possible classes of instances. *Weka* supports different data types include [24]:

- Numeric (it can be integer or real).
- String.
- Date.
- Nominal (An attribute can take one of possible values such as yes or no).

Data section contains the data instances where each line represents an instance while each column is an attribute value. The last column refers to the classification of instance.

In some cases as in text document representation, most of attributes values have values of zero. Text file usually has a small number of words in comparison with

the number of features used to represent texts in text mining. Most of the values in such case are zero. The form of data in the *ARFF* file may look like this:

{0, 0, 3, 9, 11, 0, 0, *Yes*}.

{0, 2, 0, 13, 7, 0, 0, *No*}.

In *Weka*, there is another way called *sparse ARFF* can be used for data representation. In this method, attributes are indexed starting from zero, each nonzero attribute is identified with the index of attribute. The above example can be rewritten in *ARFF* sparse file format as follows:

{2 **3**, 3 **9**, 4 **11**, *Yes*}.

{1 **2**, 3 **13**, 4 **7**, *No*}.

The first element (2 3) means the index is 2 and the attribute value is 3.

2.13 Swarm Intelligence

Swarm intelligence as defined by [51] is

> *"an artificial intelligence (AI) discipline, is concerned with the design of intelligent multi-agent systems by taking inspiration from the collective behaviour of social insects such as ants, termites, bees, and wasps, as well as from other animal societies such as flocks of birds or schools of fish".*

Many researchers have been interested in studying the collective behavior of animals and insects in an attempt to get ideas for developing sophisticated algorithms that help in solving real-world problems.

Biologically inspired approaches are based on the concept of how biological systems deal with the situation [51, 52]. The most successful biologically inspired algorithms are Ant Colony Optimization algorithm (*ACO*) and Particle Swarm Optimization algorithm (*PSO*) [51]. Researchers have categorized nature inspired techniques into five main classes: Ant Colony Optimization technique, Particle Swarm Optimization technique, artificial immune systems, membrane computing and *DNA* computing [52]. Bio inspired techniques have been successfully used to solve many

real life problems such as image processing, task scheduling, data mining, power dispatch optimization, feature selection and classification, finance applications, neural networks and medical related applications [52, 58].

2.14 Particle swarm optimization (PSO)

Particle swarm optimization was developed by Eberhart and Kennedy in 1995 [53], motivated in part by the social behaviour of flocks of birds. *PSO* is a population based stochastic optimization algorithm in which potential solutions are called particles. As well as a position in the search space (which essentially defines the solution it represents), each particle also has a velocity in the search space, which is initially random. A population of particles is randomly initialised in terms of position and velocity, and then each is evaluated; and each particle updates its velocity according to its experience and the experience of other particles in the swarm [53, 54].

Essentially, a particle will first update its velocity by moving partly in the direction of the position that has best fitness in its neighbourhood (this need not be defined geographically), and partly in the direction of the best fitness that the particle has seen in its own experience so far. The velocity thus updated, the particle itself will then adjust its position with the new velocity. Each individual (particle) is represented as a point in N dimensional space.

The particle i in the swarm is represented as $X_i = (X_{i1}, X_{i2}, \cdots, X_{iN})$ and its best previous position (position that gives best fitness value) is recorded as $P_i = (P_{i1}, P_{i2}, \cdots, P_{iN})$. The best particle in the swarm is called best global and its index is denoted by the variable g. The velocity of the particle i is represented as $V_i = (V_{i1}, V_{i2}, \cdots, V_{iN})$. Each particle in the swarm updates its velocity and position according to the following equations [53]:

$$V_{id} = \omega * V_{id} + C_1 * rand() * (P_{id} - X_{id}) + C_2 * rand() * (P_{gd} - X_{id}) \tag{2.33}$$

$$X_{id} = X_{id} + V_{id} \tag{2.34}$$

V_{id} on the right side is the previous velocity of the particle; C_1 and C_2 are two positive constants; *rand()* is random function in range [0,1]. The velocity of each particle is restricted within range [-Vmax, +Vmax] [53]. Figure 2.9 explains how particles adjust their velocity and positions according to the best particle in the swarm. In Figure 2.9, $\varphi 1 = C_1 * rand()$ and $\varphi 2 = C_2 * Rand()$.

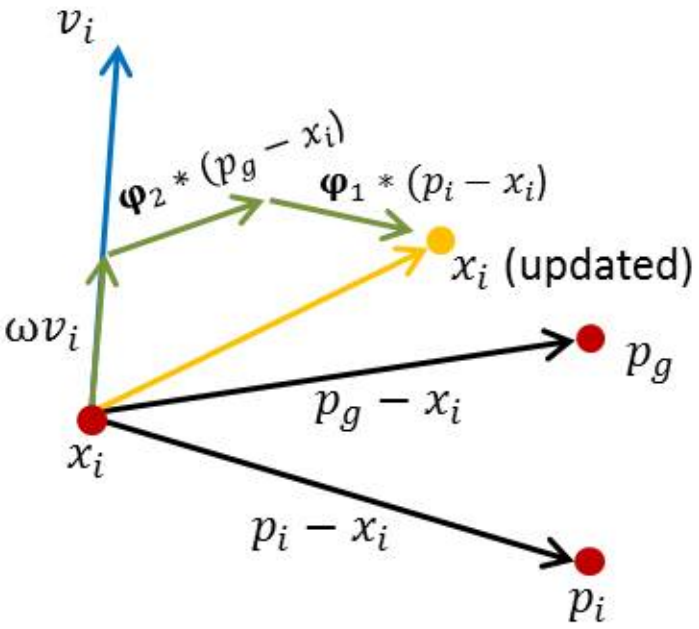

Figure 2.9: Movement strategy of the particle in PSO algorithm [51]

2.15 Variants of PSO

Since the first appearance of particle swarm optimization algorithm in 1995, many modifications have been made to the original version of *PSO*. Some researchers have proposed ideas to improve the performance of *PSO* generally while others have improved its performance for specific types of optimization problems. The modifications can be classified into three classes: extending the search space, parameter adjustments, and combining with another approach (*Hybrid PSO*) [58, 62]. In addition, *PSO* has also been used for solving multi objective optimization problems. An example of that, Hu and Eberhart proposed a dynamic neighbourhood strategy and an update of particles memory to deal with a multi objective optimization problem [63]. In this work, we will focus on the basic variants of *PSO*. Later, we will see different forms or modifications of *PSO* when we present some *PSO* applications.

2.15.1 PSO with inertia weight

In the velocity equation, the previous velocity of the particle helps particles to extend their search space (global search). Without including previous velocity, particles will fly towards the same position and then shrink the search space over the iterations (local search). Some problems benefit from both local and global search. For this reason, Shi and Eberthart introduced the inertia weight ω for balancing between local and global search [56]. The value of the inertia weight can be set as a positive constant or decreased gradually with the iterations. [56] carried out a series of experiments in order to determine the appropriate value for the inertia weight. It was found that when $\omega = 1.05$, *PSO* succeeded to find global optimum in all iterations (generation). Also, experiments showed that the inertia weight within the range [0.9 - 1.2] improves the performance of *PSO*. Using a suitable value for inertia weight, *PSO* gets a better chance to find the optimal solution in a reasonable number of generations [56].

2.15.2 Binary particle swarm optimization

The original version of *PSO* was defined for real-valued continuous search spaces; variants have since been developed that deal with discrete spaces. In binary *PSO* (*BPSO*) [55], a particles position is simply a binary vector, which initially seems difficult to reconcile with the notion of having velocities associated with a particle. Kennedy and Eberharts approach retains the equations used to manage velocities in *PSO*, with the key difference being that in *BPSO* a velocity vector (a real-valued vector in which each component is kept between 0 and 1) represents a set of probabilities, one for each component. Particle positions are realised by sampling from this vector. Meanwhile, *BPSO* is convenient and appropriate to use since binary encoding is natural for a feature selection task. The probability of the bit changing is determined by the following formula [55]:

$$S(V_{id}) = \frac{1}{1 + e^{-V_{id}}} \tag{2.35}$$

$$If(rand() < S(V_{id})) \quad then \quad X_{id} = 1;$$
$$else \quad X_{id} = 0$$

2.15.3 PSO with constriction coefficient

[59, 60] proposed a modified version of the original *PSO* algorithm that uses a constraint coefficient to control the convergence of the particles. In this regard, the velocity equation of *PSO* can be rewritten as:

$$V_{id} = K[V_{id} + C_1 * rand() * (P_{id} - X_{id}) + C_2 * Rand() * (P_{gd} - X_{id})] \tag{2.36}$$

$$K = \frac{2}{|2-\varphi-\sqrt{\varphi^2-4\varphi}|}$$

Where $\varphi = C_1 + C_2, \varphi > 4$. When this method is used, φ is usually equal to 4.1 and the constant multiplier K is 0.729 [58, 59].

2.15.4 Fully informed particle swarm optimization

In the traditional form of article swarm optimization algorithm, the movement of the particle is affected by its own flying experience (previous best) and the best global particle in the swarm (gbest). Other particles (neighbours) are not involved. In [64] a different strategy was suggested for the interaction between the particle and its neighbours called fully informed *PSO*. In this way, the particle is influenced by all its neighbours. Using *FIPSO*, the velocity equation is as follows [58, 64]:

$$V_{id} = \chi[V_{id} + \frac{1}{K_i}\sum_{n=1}^{K_i} C_1 * rand() * (P_{nbr_n} - X_{id})] \tag{2.37}$$

χ is a constraint coefficient, K_i denotes the number of neighbours of the particle (i) and nbr_n is its n_{th} neighbour. In comparison with original *PSO*, selecting appropriate parameters, the experiments showed that *FIPSO* can find a better solution with fewer generations [58].

2.15.5 Bare Bones particle swarm optimization

In Bare Bones *PSO*, Kennedy proposed a different strategy for updating a particle's position. Instead of using velocity equation to update the positions of particles, each dimension of the new position of a particle is arbitrarily generated from a Gaussian distribution with mean and standard deviation as in equations (2.38) and (2.39) respectively [123].

$$mean = \frac{P_{id} - P_{gd}}{2} \tag{2.38}$$

$$Std.Dev. = |P_{id} - P_{gd}| \tag{2.39}$$

$$X_{id} = Gaussian \quad distribution(mean, Std.Dev.) \tag{2.40}$$

2.15.6 Tracking and Optimizing Dynamic Systems

Since many applications are not always static and change their state over time, continuous re-optimization is needed. There are several solutions to deal with the problem of tracking and optimizing dynamic systems such as detection of environment change, inertia weight update, and re-initializing the swarm [61, 62]. For instance, [61] made a change to the way that inertia weight in *PSO* algorithm is set. Instead of using time decreasing to set the inertia weight, it can be selected randomly. In the velocity equation of *PSO*, the inertia weight ω was replaced with [0.5 + (Rand/2.0)]. With an average of 0.75, a random number within the range 0.5 and 1.0 is generated. Based on experimental test, it was found that *PSO* with randomly generated inertia weight demonstrated its ability to track a 10-dimensional parabolic function.

2.16 PSO topology

Two general topologies have been investigated, global best and local best. As can be seen in Figure 2.10-a, in the global best topology, all particles in the swarm are entirely connected with each other. Particles are influenced with the best particle in the swarm which can be any one that has the best fitness value. In this case, for any particle to move towards the best solution in the problem space, it needs information from all particles in the population in order to identify the best one [65, 66].

In the local best topology, there are two kinds of local best, ring and wheel. In the ring topology as shown in Figure 2.10-b, each particle communicates with only

two neighbours. In the wheel topology, all particles are entirely separated from each other and, as shown in Figure 2.10-c, the only way to communicate is through a focal particle [65, 66]. Due to its fast convergence, global best has a problem of getting trapped in local optima. Although local best has slower convergence than global best, it has a better chance of finding the best solution [65, 66]. In addition to the above mentioned topologies, another two topologies named Pyramid topology Figure 2.10-d and Von Neumann topology Figure 2.10-e were investigated by Kennedy and Mendes [65]. It was concluded that choosing the appropriate topology depends on the sort of problem.

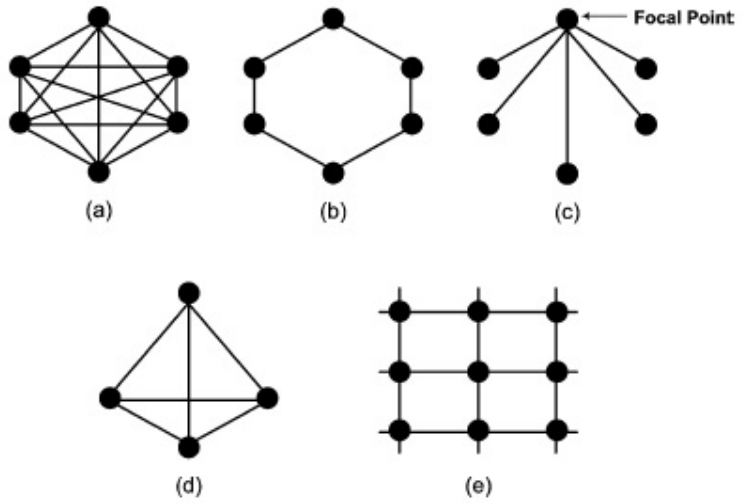

Figure 2.10: PSO topologies: (a) global best. (b) Ring topology. (c) Wheel topology (d) Pyramid topology. (e) Von Neumann (adopted from [66])

2.17 particle swarm optimization applications

PSO algorithm has been successfully applied to solve a wide variety of optimization problems. It is simple, easy to implement and has few parameters to adjust. It has also proved its effectiveness and robustness in tackling complex optimization problems. In general, similar to other evolutionary algorithms, particle swarm optimization can be used to solve a wide range of optimization problems. Some of those applications include pattern recognition, classification, signal processing, power generation, antenna design, image and video processing, neural networks, finance and economics and so forth [57, 58]. Poli, Kennedy and Blackwell [58] reviewed and

categorized over 1100 applications on the particle swarm optimization algorithm in the time between 1995 and 2006. Here, based on the survey presented in [58], we will list the number of published works in the main application areas where PSO has been successfully applied.

Area of research	*Number of publications*
Image and video analysis	51
Electricity networks and load dispatching	48
Control applications	47
Power generation and power systems	39
Electronics and electromagnetic	39
Antenna design	39
Scheduling	38
Design Applications	30
Communication networks	30
Biological, medical and pharmaceutical	30
Clustering, classification and data mining	29
Signal processing	26
Fuzzy and neuro-fuzzy systems and control	26
Robotics	23

Table 2.5: Applications of PSO algorithm [58]

As mentioned in [57], the first application that used PSO was Artificial Neural Networks (ANN). PSO was applied to select the optimal weights of the network. It was discovered that the evolved network using PSO can be used for any network structure. For instance, ANN with PSO was applied to analyse human tremor; the evolved ANN was used to distinguish between normal subjects and those that have tremor. As examples, we will present some of the recent successful work in the main areas of PSO research.

In the artificial neural networks and its related applications, PSO was applied to train a neural network to estimate the evaluation function of leaf nodes of a game tree. In comparison with the performance of a standard evolutionary method, the

results revealed that the proposed method works well in this task [67]. In [68], PSO was used to train a fuzzy neural network to extract rules for describing data. In comparison with other fuzzy neural network techniques, the results showed that similar classification rules can be extracted using the proposed method for fuzzy neural network training. In [69], PSO was applied alternatively for optimizing architecture and weights of ANN using two PSO algorithms. One algorithm is for adjusting the architecture and another one for evolving the nodes. The ANN was evaluated on the product quality estimation problem. The results show that it yielded good accuracy.

In the power systems, load dispatching and power generation field, an approach based on particle swarm optimization technique was developed to optimize the design of multi machine Power System Stabilizers (PSS). The proposed method was used to find the optimal parameter settings of power system stabilizer. Two types of PSS with different configurations and loading conditions were tested. The proposed optimization method worked well at various loading conditions and configurations. It also demonstrated its capability to damp out the electro mechanical oscillations [70].

In [71], a hybrid approach based on particle swarm optimization algorithm with arithmetic mutation was proposed to reduce the loss of power in electric power systems. Using IEEE-118 bus system for testing, experiments showed that the developed hybrid approach yielded good results. It also worked better than the original PSO for the problem of power loss reduction. In [72], a modified version of PSO was used to tackle the issue of equality and inequality constraints in Economic dispatch. In comparison with other approaches like genetic algorithm, Tabu search method, Hop-field neural network, experimental results proved that the proposed method is superior.

In the video and image analysis area, for visual tracking, to estimate the motion of an object in a given sequence of video images, a technique called sequential PSO based on the particle swarm optimization algorithm in conjunction with temporal continuity information technique was applied. In comparison with two methods, particle filter and unscented particle filter, experimental results revealed that sequential PSO approach works well for video tracking [73]. In [74], a modified version of par-

ticle swarm optimization algorithm was applied for detection and segmentation in the image analysis task. The results of different trials of the modified PSO algorithm were used for segmentation and the convergence strategy of the particles was utilized to determine and track the objects. In terms of real time performance, the suggested method was shown to be effective.

In [75], PSO was used to select the best subset of features for the face recognition task. The discrete cosine transform (DCT) and the discrete wavelet transform (DWT) techniques were applied to extract vectors of features from the face recognition database. The fitness function of PSO was defined by the best class separation. In comparison with the FS method based on GA, PSO based feature selection approach yielded similar results with less features.

In the area of multi robot systems, a group of robots interact with each other to reach their goal. One of the major problems in this field is how to develop algorithms that can help robots to achieve their goal efficiently. For optimizing multi robots interaction, a modified particle swarm optimization algorithm was used to simulate the search procedure of multi robot systems. Using various numbers of robots and communication range, the developed search algorithm managed to find its target efficiently [76]. In [77], two PSO swarms were applied to the collective robot search task. An inner swarm was used to find the target while an outer swarm was utilized to select the optimal values of the quality factors, inertia weight, C_1 and C_2 of the velocity equation for the inner swarm. Results showed that the PSO algorithm is capable of finding the target in both cases of search, single and multi target.

In data mining, pattern recognition and classification, feature selection is a crucial process that aims to improve the classification accuracy and computational efficiency by removing irrelevant and redundant terms while retaining features that contain sufficient information to assist with the classification task at hand. Particle swarm optimization has been used to solve this problem. For instance, in [78], a new feature selection technique based on particle swarm optimization algorithm and rough sets was proposed. Using UCI data, in comparison with other approaches such as GA-based FS approach and rough set features reduction approaches, experimental results revealed that PSO with rough sets works well for feature selection.

[79] suggested a feature selection technique based on an improved version of the binary particle swarm optimization with support vector machine. The goal of this technique is to overcome the falling of particles into local optima by adding what is called adaptive mutation. In this way, particles that fall into local optima are re-initialized again. In contrast with the classification accuracy of other methods such as GA-SVM, and SVM without feature selection, IBPSO-SVM is superior. It obtained the highest accuracy with the lowest number of features. Also, in [80] BPSO with logistic map was used to serve as an FS technique. Logistic map approach was applied the find the appropriate value for the inertia weight. In addition K nearest neighbour classifier was used for evaluating the performance for selected subsets of features by BPSO. This method achieved better accuracy than other techniques such as sequential forward search, sequential genetic algorithm and hybrid genetic algorithm.

In [81], a combination of Binary PSO and estimation distribution algorithms EDA was adopted as a feature selection technique for searching the best subset of single nucleotide polymorphisms (SNPs) for Crohns disease and lung cancer. SVM was used as a classification model. The experimental results revealed that the performance of BPSO with EDA is higher than other techniques that include higher selection pressures compact GA with SVM, BPSO with SVM and SVM without FS method. [82] proposed a feature selection method based on Binary PSO and overlap information entropy (OIE). The idea is to use the degree of correlation between features and categories calculated by OIE to evaluate the subsets of features selected by the particles. For testing, homogeneous protein datasets were used. KNN was applied for evaluating the best subset of features selected by BPSO with OIE; experimental results showed that this method works well for the feature selection problem.

As described in [83], an improved version of BPSO was used to serve as a feature selection method for gene expression data classification. The idea is to reset the global best (gbest) if it does not change after three generations. In addition, KNN classifier was used to evaluate the selected features by IBPSO. Using eleven gene expression datasets, the classification accuracy obtained by IBPSO method was the best on nine datatests.

2.18 Arabic Text Classification

2.18.1 Arabic language

Arabic has 28 letters and is written from right to left. In contrast with English, Arabic has a richer morphology that makes developing automatic processing systems for it a highly challenging task. The basic nature of the language, in the context of text classification, is similar to English in that we can hope to rely on the frequency distributions of content terms to underpin the development of automatic text classification. However, the large degree of inflections, word gender, and pluralities (Arabic has forms for singular, dual, and plural), means the pre-processing (e.g. stemming) stage is more complex than in the English case [27,84, 87].

Arabic language has three genders, feminine, masculine and neuter [128]. In general, Arabic words are classified into three main groups; nouns, verbs, and particles. Noun in Arabic is defined as a word that describes person, thing, place or idea [127]. Nouns in Arabic can be derived from other nouns, verbs, or particles [127]. Verbs in Arabic are divided into perfect, imperfect and imperative. Arabic particle category includes pronouns, adjectives, adverbs, conjunctions, prepositions, interjections and interrogatives [127]. Based on fixed patterns called "Awzan", most of Arabic words can be obtained from stem or root of words by attaching prefixes, suffixes and infixes to the root of word [128, 131]. Arabic roots are composed of three, four, or, in some cases, five letters [130].

In contrast with phonetic symbols in English, Arabic language has a set of diacritics which are used to pronounce words correctly. Diacritic marks can be written below or above letters. They are short vowel marks. The main Arabic diacritics include Fatha, Dama, Kasra, Shada, Sukun and Tanween [118, 129, 130]. For instance, Table 2.6 presents different pronunciations of the letter (Sean) (س) [118].

سَ	سِ	سُ	سْ	سَّ	سِّ	سُّ	سّ
/sa/	/si/	/su/	/s/	/ssa/	/ssi/	/ssu/	/ss/

Table 2.6: Different pronunciations of the letter (*Sean*)

The difficulty of Arabic natural language processing in general and Arabic text

classification in particular is related to the nature of Arabic language. Here, in comparison with other languages such as English, we list some of aspects that make automatically processing Arabic language a challenge task [126, 129, 130, 132]:

- Arabic language has a complex morphology in comparison with English. An Arabic word is usually built up from a root attached with affixes. As an example, Table 2.7 presents different morphological forms of word study (دراسة) [129].

Word	Tense	Pluralities	Meaning	Gender
درس	Past	Single	He studied	Masculine
درست	Past	Single	She studied	Feminine
يدرس	Present	Single	He studies	Masculine
تدرس	Present	Single	She studies	Feminine
درسا	Past	Dual	They studied	Masculine
درستا	Past	Dual	They studied	Feminine
يدرسان	Present	Dual	They study	Masculine
تدرسان	Present	Dual	They study	Feminine
درسوا	Past	Plural	They studied	Masculine
درسن	Past	Plural	They studied	Feminine
يدرسون	Present	Plural	They study	Masculine
يدرسن	Present	Plural	They study	Feminine
سيدرس	Future	Single	He will study	Masculine
ستدرس	Future	Single	She will study	Feminine
سيدرسا	Future	Dual	They will study	Masculine
ستدرسا	Future	Dual	They will study	Feminine
سيدرسون	Future	Plural	They will study	Masculine
سيدرسن	Future	Plural	They will study	Feminine
يدرسا	Present	Dual	They study	Masculine
تدرسا	Present	Dual	They study	Feminine

Table 2.7: Different morphological forms of word (*Darasa*)

- In English, words are usually composed of a root attached with prefixes and/or

suffixes. In Arabic, infixes can be added inside the word. For instance, in English, the word *write*is the root of word *writer*. In Arabic, the word writer(كاتب) is formed differently from English. It is formed by adding the letter Alef (ا) inside the root (كتب). In such cases, especially in process like stemming, it is difficult to distinguish between the root letters and infixes [126].

- Semantic, morphology, and syntactic of Arabic language is different from, more complex than Indo-European languages [132].

- Some Arabic words may have different meanings depending on their appearance in the context. Especially in Arabic scripts in digital form, mostly, diacritics are not used, the proper meaning of the Arabic word can be determined based on the context. For instance, the word (ذهب) could be noun *gold* (ذَهَبْ) or verb *went* (ذَهَبَ) depending on the context [130].

- In Arabic language, Irregular plurals and synonyms are widespread [101, 129].

- An example of the challenges of Arabic text automatic processing is the problem of dealing with proper nouns, since Arabic letters do not have lower and upper case, proper nouns in Arabic do not begin with capital letters as in English; the process of capturing such words in Arabic text is more difficult than in English [126].

- For Arabic TC, Arabic corpus with its precise training and testing portions is not publically available for research purpose. This makes the comparison between Arabic TC approaches not possible [129, 132]. In this work, to overcome this issue, we have formed three Arabic datasets (each dataset is split into training/test portions) and, made them available for other researchers to directly compare with our results.

- Arabic language has special encoding. The use of unsuitable encoding will result in improper Arabic text display. The most common used Arabic text encodings are UTF and CP-1256 Arabic windows [129].

2.18.2 Related work

Limited work in the area of TC has been done so far for Arabic text documents. Although several studies have been reported, this area is still at an early stage. For instance, although publically available Arabic text datasets exist, it is very limited for any such dataset to be used in more than one work. In the following, we review what has been done, and we also quote reported classification accuracies; however these present a broad idea of performance, and almost never show performance on a common dataset under common conditions. The most common previous work in the area of Arabic TC includes the following.

In [86], three experiments have been preformed to classify Arabic documents using K Nearest Neighbour classifier. Different similarity measures including Cosine, Jaccard and Dice were tested. In addition, different Arabic datasets gathered from online Arabic newspapers were used in these experiments where each dataset consists of six classes: Agriculture, Art, Economy, Health, Politics and Science. For term weighting, different approaches including TF.IDF, WIDF, ITF and log (1+TF) were tried. In addition, each dataset was split into two portions, 70% for training and 30% for testing. The K parameter in KNN was set to 11. Using F1-measure, the results show that the highest accuracy was scored by Cosine technique with log (1+TF), it was 92.64%; while Dice and Jaccard achieved an accuracy average between 81.01% and 94.91% with the four term weighting methods.

[87] evaluated the performance of two well known classification algorithms C5.0 and Support Vector Machine on classifying Arabic text documents. Seven different Arabic datasets were used. Each dataset was divided into 70% training set and 30% test set. Chi square approach was used to select the 30 most relevant terms in each class for each data set. [87] reported that the average accuracy of SVM was 68.65% while C5.0 (Decision Trees) outperformed with average accuracy of 78.42%.

[88] implemented an Arabic classification system called ArabCat. This system uses maximum entropy method to classify Arabic texts into pre-defined groups based on their content. An Arabic dataset divided into six domains was collected from the internet especially from the web site of Aljazeera Arabic news channel for evalua-

tion. Also, pre-processing steps including stemming and stop words removing were done. Without performing pre-processing, F1-measure reached 68.13% and with the normalization only, the performance increased to 70.25%. In the normalization stage, the texts were converted into UTF-8 encoding. In addition, punctuation and non-letters were removed. By using both normalization and generating all possible forms of the terms using tokenizer, the performance increased to 71.20%. [88] also reported that the system was tested using only nouns and pronouns as features while excluding all other words. The performance increased to 80.41% and this accuracy was the highest.

[89] used Naive Bayes algorithm to classify Arabic web pages into pre-defined categories. The used data set was divided into five categories. Each category consists of 300 web documents. The performance of NB was tested using a cross-validation technique. The experiments showed that the classification accuracy over all categories was 67.78%.

SVMs were used in [90], in conjunction with a chi square feature selection method for selecting the most relevant features for each category to classify Arabic texts. An Arabic dataset consisting of 1445 collected from online Arabic newspapers archives was used for training and testing the Arabic TC model. Also, TF.IDF scheme was used for term weighting. Experimental results showed that the average accuracy over all categories using F-measure was 88.11%.

In [91], a probabilistic classification system called Barq has been implemented and used to classify Arabic texts into 12 pre-defined groups. An Arabic dataset consisting of 1000 Arabic documents was used for training and testing the Arabic TC system. Also, a cross-validation method was performed for performance evaluation. The results showed that the overall performance accuracy was up to 75.6%.

According to [92], Naive Bayes classifier based on Chi square feature selection technique was applied to classify Arabic texts based on their content. The dataset used here was collected from the Saudi Press Agency website. It consists of 1562 Arabic documents which fall into six categories: Economic, Cultural, Political, Social, Sport, and General. The experiments showed that NB classifier for Arabic texts

classification worked well and for Arabic TC. In terms of F1 measure, the highest performance accuracy was between 72.5% and 73% when the number of the selected features was between 800-1000.

In [93], comparison was made between three classifiers including Distance Based classifier, K-Nearest Neighbour and Naive Bayes to classify Arabic texts. An in-house collected Arabic dataset was used to estimate the performance of those classifiers. The dataset consists of 1000 Arabic documents, categorized into ten classes (100 documents per class). Also, pre-processing includes word stemming and stop words removing was conducted in order to reduce the dimension of the feature vector space. The experiments showed that NB classifier has achieved the best classification accuracy in comparison with the other two classifiers using the same Arabic dataset.

N-Gram frequency statistical technique was used in [94] to classify Arabic documents. An Arabic dataset collected from online Arabic newspapers was used. The data set covers four categories: sports, economy, technology and weather. Also, two similarity measures were applied, one is Manhattan distance measure and the other is Dice measure. The results show that the best accuracy was obtained using Dice measure in conjunction with tri-gram frequency method.

In [95], Support Vector Machines algorithm with different feature selection methods was used to classify Arabic texts into pre-defined categories based on their content. Arabic dataset consisting of 1445 Arabic texts collected from online Arabic newspapers was used. In addition, a pre-processing including Arabic letters normalization, non Arabic words removal, stop words elimination was conducted. Also, TF.IDF was applied for term weighting. Six feature selection methods were used to select the most relevant features for each class in the dataset. These methods include: Chi square, Information gain, Mutual information, NGL (Ng-Goh-Low) Coefficient, OR (Odd Ratio), and GSS (Galavotti-Sebastiani-Simi) Coefficien. Three different experiments were conducted using 140,160 and 180 top features. The results show that CHI, NGL and GSS performed well for the Arabic TC task. The macro average of F1-measure using those three FS methods was over 80%.

[96] investigated the performance of well-known ML algorithms CBA,Naive Bayes

and SVM on classifying Arabic text documents into seven pre-defined classes. The dataset used for the investigation was obtained from Saudi online newspapers and it consists of 5121 Arabic documents. F1-measure was used to evaluate the performance of the three algorithms. The results show that CBA outperformed NB and SVM and the average of its F1-measure is 0.804.

In [97], KNN classifier was tested for classifying six hundred Arabic documents relating to six pre-defined categories. The Arabic dataset was pre-processed by removing the stop words and conducting light stemming. Also, information gain technique has been used for filtering the most promising features for the classification. Also, training sets with different sizes were tested. Experimental results showed that using Jaccard similarity measure the macro recall reached 0.793 and the macro precision was about 0.627. The authors also reported that increasing the size of the training set improves the performance of KNN.

In [98], N-gram frequency statistical technique has been used to classify Arabic text documents into four pre-defined classes. The Arabic dataset used in the experiments was collected from four online Jordanian Arabic newspapers archives. Pre-processing includes stop words and non Arabic letters removal. Documents were represented as vectors of tri-grams. The classification was achieved using similarity measures Dice and Manhattan. The classification is done by calculating the distance between the unseen instance and all documents in the training; then, the unseen instance is assigned to the category with the smallest calculated measure. Results showed that text classification using Dice measure leads to better performance than the classification using Manhattan measure.

In [27], the K-nearest neighbour classifier was used to classify fifteen thousand Arabic texts based on their content into three pre-defined categories. Pre-processing steps includes stop words and non Arabic letters removal. Two experiments using KNN classifier were tried to investigate the effect of stemming and light stemming on the classification accuracy, one after performing stemming and another after performing what is called light stemming. In the stemming process, words are reduced to their 3-letter roots while lighting stemming involved eliminating the frequently used suffixes and prefixes in Arabic. In addition, term frequency TF method was

used for feature weighting. The results showed that light stemming performed better than stemming. The micro average of precision using light stemming was 0.91 while using stemming was 0.87.

[101] investigated the effect of using combinations of various term weighing schemes on the Arabic document classification problem. C4.5 decision tree was used for classification. An Arabic dataset gathered from Aljazeera News website was used for building and testing the classifier. In terms of performance and classification accuracy, the results show that the combinations *wc-norm* (word count with normalization) and *wc-tfidf* (word count with TF.IDF) worked well for Arabic TC. The proposed Arabic TC model was evaluated on a dataset contains 119 files. The best classification accuracy with 10-fold-cross validation is 97.4. It was obtained using *wc-norm-minFreq3* for stemmed words.

[102] studied the performance of two well-known machine learning algorithms KNN and SVM on classifying Arabic texts. Chi square method was used as a feature scoring method and TF.IDF was used for features weighting. A collection of Arabic news articles belonging to two categories and divided into training set and test set was used to evaluate the performance of the classifiers. The results proved that both classifiers worked well in the Arabic TC task. In terms of prediction time and F1-measure, SVM was better than KNN classifier.

[103] compared the performance of SVM and nave Bayes classifiers on classifying Arabic documents. A set of 2244 Arabic text documents (Islamic articles) relating to five categories was used in the experiments. In terms of F1-measure, the results show that SVM was superior to NB. The classification performance of SVM was 0.954 while NB was 0.884.

[105] conducted a similar study to compare the performance of SVM and NB on classifying 5121 Arabic documents belonging to seven categories. Experiments revealed that SVM is superior to NB. Using F1-measure, the average accuracy of SVM was 0.778 while NB was 0.74.

[104] made a similar comparison as [103,105] between SVM and multinomial NB

but on a different Arabic dataset consisting of 7034 texts belonging to seven classes. Also, Chi square and Information gain feature selection methods were applied. In addition four feature types were tested. Those types cover: character level n-grams, words as they appear in text (without pre-processing), lemmas and roots. Different numbers of features ranging between 400 and 2000 were tested. Using stratified cross validation, experiments showed that the best result in terms of F-measure was scored by SVM with 3-grams feature which was about 0.92. It was noticed that there was no significant difference between using IG and Chi square for feature selection.

In [106], the author evaluated the rule based algorithms One Rule, rule induction (RIPPER), C4.5 and hybrid PART to classify Arabic documents. The dataset used here is a set of 1526 Arabic documents related to six classes, collected from an Arabic website. The Chi square method was applied to filter the 30 most relevant features to be used for text representation. The results showed that the hybrid approach PART outperformed the other three algorithms. Using a different Arabic dataset called CCA, [108] investigated the performance of the same rule based algorithms used in [106]. Based on the obtained results, the authors concluded that C4.5 is the most appropriate classifier for Arabic TC problem in comparison with the other three algorithms.

In [107], decision trees classifier was applied to classify Arabic textual data. Term frequency threshold with the embedded information gain of decision tree was proposed as a feature selection method. Also, two Arabic datasets were used for testing purposes. One is a set of scientific Arabic articles. It consists of 373 documents belonging to 8 categories. The second one is a set of 453 Islamic articles distributed over 14 categories. One third of each dataset was held for testing and the remaining portion was used to build the classification model. Results revealed that the suggested method is very effective. The accuracy over the dataset of scientific Arabic articles was 0.93 whereas the accuracy reached 0.91 using the Islamic articles dataset.

[109] investigated the performance of Support Vector Machine classifier on classifying Arabic texts. Seventeen different feature scoring metrics were examined. In addition, a corpus of Arabic texts containing 7842 documents distributed over ten categories was used for evaluation. For the Arabic TC problem, results indicated

that fallout and Chi square feature selection achieved better results than all other tested metrics.

[110] investigated the effectiveness of using KNN classifier in conjunction with different text representation methods for Arabic TC task. The applied text representations include N-gram, bag of words and what [110] called conceptual representation. The new proposed reorientation was built by replacing words (terms) with their suitable concepts using an Arabic WordNet resource. Also, Chi square approach was used for feature selection. The Arabic dataset used for constructing and testing the classifier was obtained from [90]. In addition, three distance measures were used with KNN for comparison purposes. Results revealed that the best F-measure was 0.74. It was obtained when the conceptual method and KNN with Cosine distance measure were applied.

In [26, 27, 28], Arabic light stemming algorithms were developed to be used as feature selection techniques to improve Arabic text classification. [27] proposed a light stemmer called local stem ETS2. To find the local stem of a word in a specific text, the local stem builds a list of all possible syntactically related words to that word in the given text, and then picks the shortest form of related words as the local stem of the word. For testing the local stemmer, a dataset consisting of 2966 Arabic documents distributed over three categories was used; three well-known algorithms NB, SVM and KNN were used for classification. Also, another two Arabic stemmers were used for comparison purposes. Using the cross validation approach, the results of T-test and ANOVA showed that the proposed stemmer is superior to the other stemmers. [27] compared the effect of using two Arabic stemmers in Arabic TC. Stemmer and light stemmer were applied. Stemmer reduces words to their stem or root while light stemmer strips some prefixes and suffixes of words. An Arabic corpus consisting of 15000 texts belonging to three categories was used in the experiments. Also, KNN algorithm was applied. Results showed that light stemming works better than stemming in Arabic TC. [29] implemented an Arabic light stemmer to be used for the Arabic TC task. The stemmer was tested using 400 Arabic documents related to four classes and divided into training and testing sets. In terms of precision and recall, results show that Arabic light stemming improves the performance of Arabic text classification.

For the Arabic text classification problem, [111] proposed a feature subset selection method based on the Ant Colony Optimization algorithm. Chi square was used with ACO as heuristic information criteria. Also, SVM was applied to guide the search for a good set of features to be used for Arabic TC. The Arabic dataset used is the same used in [95]. It consists of 1445 documents distributed over nine categories, divided into training and test sets. The number of documents in the training set is 966 while in the test set it is 479. The first part of this work involved conducting a comparison between a set of filter approaches for the Arabic TC task. Using SVM as a classifier, it was found that Chi square is the best filter method for this task. The Macro averaging F-measure reached 87.54. After that, the proposed FS method ACO was applied and the macro averaging F-measure increased to 89.16.

Finally, according to [99], a feature selection algorithm based on the evolutionary algorithm Particle Swarm Optimization (PSO) in conjunction with Radial Basis Function (RBF) networks technique was used as feature selection method to improve the performance of the Arabic text classification. An Arabic dataset collected from online Arabic newspapers archives was used for evaluation purposes. The dataset consists of 5183 Arabic text documents categorized into ten pre-defined classes. Experimental results show that the proposed FS method is superior in comparison with the performance of the statistical feature selection methods TF.IDF and Chi-square. In terms of F1-measure, the performance of the POS based FS algorithm reached 93.9% while the performances of TF.IDF and Chi-square methods were 92.1% and 85.5% respectively.

2.19 Summary

In this area, the key concepts related to text classification were explored. The chapter explained the text mining field and its importance. This chapter also defined the text classification problem and viewed its task in organizing large volumes of text documents to help in searching and/or browsing large collections of documents.

The main steps of text classification process were also presented (text pre-

processing, feature selection and classification model construction). For text pre-processing, the main steps of text pre-processing were discussed. In addition, the general types of feature selection (filter and wrapper) were also presented. Moreover, the most common machine learning classifiers which are usually applied to TC problem were also discussed. This chapter also introduced weka machine learning tool and briefly mentioned its powerful tools for data mining such as classification, clustering, data pre-processing, visualization and, data sources.

Because this thesis suggests a wrapper feature selection approach based on Binary Particle Swarm Optimization algorithm for Arabic document classification, this chapter discussed the PSO algorithm and showed its capability in tackling many optimization problems. It also surveyed its variants and successful applications in different areas such as electric power systems, image processing, robotics and medical applications.

In addition, the chapter also briefly presented the Arabic language and showed some of its differences in contrast with the English language. In Arabic text classification field, the author reviewed what has been done, and also quote reported accuracy figures that signal a very broad idea of performance. Since common datasets have not used as noticed form reviewed Arabic TC work. The results in Arabic TC area almost never represent performance on a common dataset under common conditions.

Chapter 3

Arabic Text Pre-processing

This chapter provides a description of Arabic datasets used in experiments for testing our proposed feature selection method for Arabic TC problem. It illustrates the steps of pre-processing Arabic documents and how those documents can be converted into TF.IDF term vectors. It also reports the number of distinct features in the training portion for each dataset separately.

3.1 Arabic Datasets

Three separate Arabic datasets have been used in this research to test the proposed features selection method. Each is taken from a different previous paper in this area, and we use all or part of it in our experiments. We note again here that direct comparison with previous work, despite the dataset availability in some cases, is compromised by the fact that in these cases it has been difficult to clarify the precise ways in which datasets were organized into training and test sets, and/or how the results in the published papers relate to training or test data. In our case, we provide full details below and the associated datasets with clear partition into training and test data, here: [112].

3.1.1 Akhbar-Alkhaleej Arabic Dataset

The Akhbar-Alkhaleej Arabic Dataset is a collection of 5690 Arabic news documents gathered evenly from the online newspaper "Akhbar-Alkhaleej". It is available from

[113] and an example of research using it is [114]. It consists of four categories and each document in this collection has only one category label (single-labeled). In this work, we have selected 1708 documents randomly. Table 3.1 shows the distribution of the selected documents among the four categories.

Category	Train	Test	Total
International News	228	58	286
Local news	576	144	720
Sport	343	86	429
Economy	218	55	273
Total	1365	343	**1708**

Table 3.1: *Akhbar-Alkhaleej Arabic Dataset*

3.1.2 Alwatan Arabic Dataset

The Alwatan Arabic Dataset is a collection of 20,291 Arabic news documents gathered evenly from the online newspaper "Alwatan" [115]. It consists of six categories where each document in this collection has only one category label. In this work, we have selected 1173 documents from four categories randomly. Table 3.2 shows the distribution of the documents among these four categories. This corpus is available online at [113].

Category	Train	Test	Total
Culture	156	67	223
Religion	216	93	309
Sport	255	109	364
Economy	194	83	277
Total	821	352	**1173**

Table 3.2: *Alwatan Arabic Dataset*

3.1.3 Al-jazeera-News Arabic Dataset

The Al-jazeera-News Arabic Dataset (Alj-News) is an Arabic dataset obtained from [84]. This dataset consists of 1500 documents. It includes five categories (Sport,

Economy, Science, Politics and Art). The number of documents in each category is 300 documents. The size of the training set is 1200 documents (240 texts for each category), and the size of the test set is 300 documents (60 texts for each category). This dataset is available at [116].

3.2 Text pre-processing

All Arabic text documents have been pre-processed according to the following steps [98, 24, 25]:

- Conversion to UTF-8 encoding.
- Remove hyphens, punctuation marks, numbers, digits, non-Arabic letters and diacritics.
- Remove stop words.
- Eliminate rare words (words that occur less than five times in the dataset).
- The standard Vector Space Model (VSM) was used to represent Arabic texts [4] and TFIDF was used for the term weighting factors.

Diacritics in Arabic have a similar function as phonetics in English. In comparison with English, the difference is that Arabic diacritics are written with words [118, 121]. In modern writing of Arabic texts such as news and scientific articles, diacritics are not used [122]. In this work, during tokenization of Arabic documents, diacritics are removed to avoid the repeating of the same words with various writing forms e.g. with and without diacritics which may result in higher feature space, and affect the efficiency of our Arabic TC technique.

In stop words removal, words that appear frequently in texts and have little or no information that help retrieval efficiency in IR systems or discriminating between categories in TC are called stop words. They are usually eliminated. Arabic stop words are similar to those in English such as then, in, or, that, they, go, wait, entirely and so forth. In this work, a stop word list was created for eliminating Arabic stop words. Possible stop words from Arabic word categories such as pronouns, prepositions, verbs and letters have been collected and used to form a list of stop

words to be used in Arabic documents pre-processing [119,120, 124].

The pseudo code for extracting the distinct features from a given training corpus is as follows:

```
Define List_of_words_each_file<word,count>
Define List_of_Distinct_features<word,count>

for each text_file do
{
Temp_List=Tokinze()
for each token Temp_List do
 {
if(!Check_Stop_Words(stop_word_list,token)) then
  {
if(!File_Check(token,List_words_each_file[])
List_words_each_file[].add(token)
else
List_words_each_file[].Increment(token)
  }
 }
}
for each List_words_each_file do
{
for each token in List_words_each_file do
 {
if(!Check_Word(List_of_Distinct_features,token)) then
  {
Add_Word_Count(List_of_Distinct_features,Word,count)
else
Add_count_only(List_of_Distinct_features,Word,count)
  }
 }
}
```

The output of the above procedure is a list of distinct words (features) with their counts. The next step is to remove rare words (words that appear few times, usually less than pre-defined threshold in the corpus, are ignored).

After rare words elimination, the last step is to calculate the TFIDF of each distinct feature for each document in the corpus. In this way, each text file in the dataset is represented as a vector of features (words). In this way, each text document in the collection is represented as a vector of features i.e. $(f_{i1}, f_{i2}, \cdots, f_{iN})$ where i refers to the i_{th} document in the dataset and N is the number of features that represent the documents. After pre-processing the three Arabic datasets, the numbers of distinct features found in the separate datasets were as shown in Table 3.3. We will see later that these directly represent the sizes of the Binary PSO vectors (particles). Hence, each text document in each dataset was transformed into a feature vector.

Dataset	*Distinct features from the training set*
Alj-News	5329
Alwatan	12282
Akhbar-Alkhaleej	8913

Table 3.3: Number of distinct features in the training portions of the three datasets

The developed pre-processing software (using Java Eclipse Plateform) generates training set based on all distinct features to be used by the feature selection approach (BPSO) to select the best subset of features. Then, the training set is rebuilt based on the best subset of features selected by BPSO FS approach; the test set is built using only the best subset of features. In all our experiments, the pre-processing and weighting were performed separately for the training set and the test set. That is, for example, TFIDF weightings for the test set were not influenced at all by the training set.

Based on the best subset of features generated using feature selection techniques we will see later, the developed java code for Arabic text pre-processing generates two datasets (training and test) in Weka data file format (ARFF). Figure 3.1 shows the format of the training portion of the Alwatan dataset. The file was viewed using

Weka Dataset file (ARFF) Viewer.

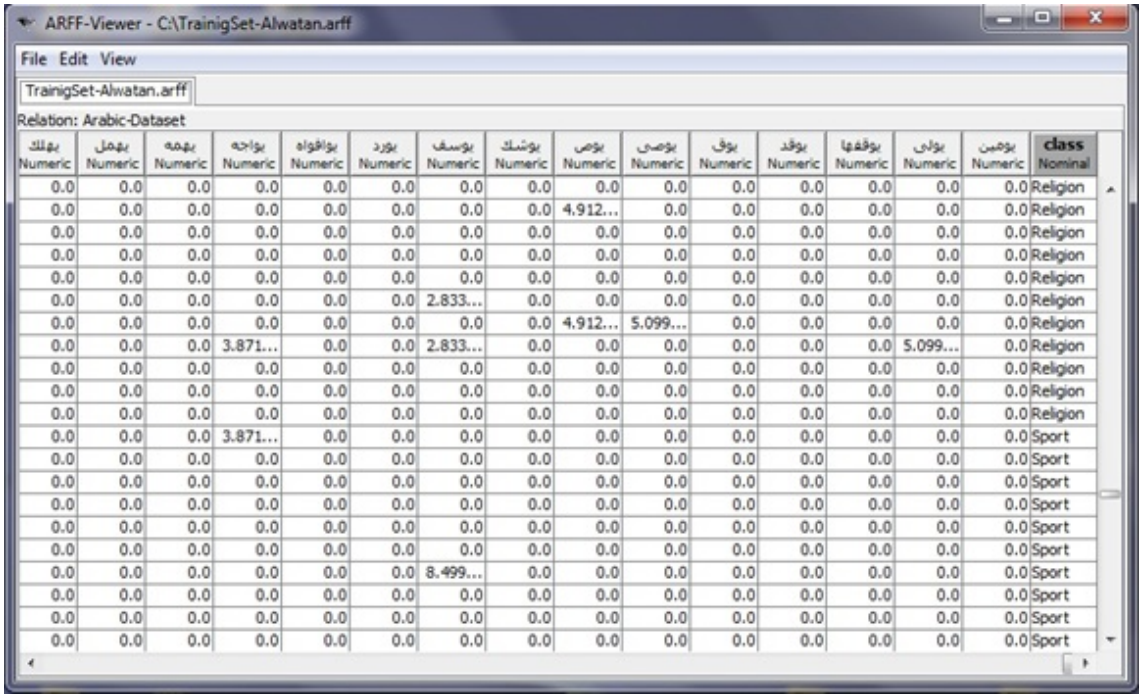

Figure 3.1: View ARFF file in Weka

3.3 Summary

This chapter presented the three Arabic datasets used in this thesis where each dataset was shown with its precise number of documents in its training/test portions. It also illustrated the pre-processing steps of Arabic text documents and, showed how Arabic documents have been transformed into TFIDF feature vectors. In addition, the number of distinct features from each dataset was also reported.

Chapter 4

Feature Subset selection for Arabic TC using BPSO with K Nearest Neighbour

In this chapter, we demonstrate a combination of Binary PSO and K nearest neighbour that performs well in selecting good sets of features for Arabic text document categorization. On each training set of the three Arabic datasets described in the previous chapter, BPSO/KNN was run ten times. The selected subsets features, in conjunction with SVM, Naive Bayes and C4.5 classifiers were used to classify the hold out test sets. The overall summarised results on the test set in terms of F-measure (recall this is an average of 10 complete trials of the training/test process) show that BPSO/KNN works well for selecting good sets of features for Arabic TC task. The effect of conducting normalization and Arabic light stemming was also investigated. The same number of experiments on the three datasets with the three ML classifiers was done and the results were compared with the results obtained without applying normalization and light stemming.

4.1 BPSO/KNN feature selection method

The approach we propose and test in the next section is aimed at finding a good subset of features to support the task of Arabic text categorisation. Comparisons with other techniques are made with the help of the weka machine learning library

[50, 117]. Note that we clearly distinguish the task of feature selection from the question of classification. That is, we use a BPSO/KNN hybrid method, working on a training set, to output a specific subset of features. We then evaluate this subset of features on a test set, in which we can use any machine learning/classification method for the evaluation. In fact we evaluate feature subsets using Naive Bayes, J48 (Wekas implementation of C4.5) and an SVM with a linear kernel (using Weka machine learning software version 3.6.3) [49, 50]. In this section, we describe our feature selection method, which is a BPSO/KNN hybrid. A step by step view is given below; this all follows a text pre-processing step, described in the previous chapter, in which a total of N terms (features) are pre-determined from the document collection.

Step (1): Create a population of particles on N dimensions in the feature space. Each particle is represented by three vectors: the particles current position (X_i), the particles best previous position (P_i) and its velocity (V_i). X_i is initialized with random binary values where 1 means the corresponding feature is selected and 0 indicates not selected. P_i is initialized with a copy of X_i. (Following evaluation of each particle in the swarm, the global gbest is initialized with the index of the particle with best fitness value).

Step (2): For each particle:

- Evaluate fitness using KNN classifier (see below).
- Update particles personal best.

Step (3): Update global best gbest.

Step (4): Update velocity and position of all particles in the population according to standard approach in BPSO [55].

$$V_{id} = \omega * V_{id} + C_1 * rand() * (P_{id} - X_{id}) + C_2 * rand() * (P_{gd} - X_{id}) \tag{4.1}$$

The probability of the bit changing is determined by the following formula [55]:

$$S(V_{id}) = \frac{1}{1 + e^{-V_{id}}} \tag{4.2}$$

$$If(rand() < S(V_{id})) \quad then \quad X_{id} = 1;$$
$$else \quad X_{id} = 0$$

Step (5): Terminate if termination criterion is satisfied, outputting the selected subset of features (represented by the current global best particle), else go to step (2).

The fitness of a particle is calculated using the following formula [99]:

$$Fitness = (\propto * Acc) + (\beta * (\frac{N - T}{N})) \tag{4.3}$$

Where

- Acc is the classification accuracy of the particle found using KNN (see below).
- $\propto$ and β are two parameters used to balance between classification accuracy and feature subset size (selected by particles), $\propto$ is in the range [0,1] and β = 1 - $\propto$.
- N is the total number of features.
- T is the length of the selected subset of features by particle.

The classification accuracy of a particle (P) is calculated using the following procedure:

- Filter the subset of features selected by P.
- Set C=0.
- For each instance in the training set (this is during the training phase; all results presented in this thesis are based on unseen test data).
    - Calculate the Euclidean distance from the current instance to all instances in the training set.
    - Classify the current instance according to its K nearest neighbours in the training set.
    - If the predicted classification matches the known classification of the instance, increase C by 1.
- Finally the Classification accuracy of P is recorded as C divided by the total number of instances in the training set.

4.2 BPSO/KNN parameter settings

In order to find out appropriate parameters of the BPSO algorithm, experiments were conducted on 150 documents falling in 5 categories (each category has 30 files): sport, science, art, politics and economics. Those documents were selected randomly from the training portion of *Alj-news* dataset. Experiments of BPSO concentrated on parameters including:

- Inertia weight ω.
- Number of generations.
- K parameter for KNN classifier.
- Swarm size.

4.2.1 Inertia weight

The aim of introducing inertia weight is for balancing the local search and the global search [56]. As mentioned in [56], the best range of search to find suitable value for inertia weight (ω) is between 0.9 and 1.2. We have tried two ranges to find out the best value of ω, the first range is between 0.4 and 0.9 with 0.1 scales. The second one is between 0.9 and 1.2 with 0.01 scales. Figure 4.1 shows the obtained results. Each point represents an average of 10 complete trials of best fitness value for each tested value of inertia weight (ω). In all performed experiments, BPSO/KNN has the following parameters:

- No of iterations (generations): 100.
- Swarm size: 30.
- K parameter of KNN classifier: 3.
- Rare words: words that appear less than 5 times in training set were removed.

Three values were tried for the parameter in the fitness function (0.7, 0.85 and 1). Based on the experimental results, the best value of inertia weight is 1.02. When $\propto$=1, β=0, this means no consideration to length of selected subset of features by BPSO particle. To balance between the classification accuracy of particle and the

length of selected subset of features by that particle we assume that classification accuracy is more important than subset length, so $\propto$ equals 0.85 was chosen.

Each point in Figure 4.2 shows the number of times BPSO succeeded in finding global best out of 100 iterations (each point is an average of 10 runs). The highest averages were recorded when inertia weight is 1.02 or 1.01. Table 4.1 shows the average best fitness values with inertia weight equals 1.02 or 1.01 and Table 4.2 represents the average number of times global best was found with best values of (ω).

Inertia weight (ω)	$\propto$=1	$\propto$=0.85	$\propto$=0.7
1.01	0.92	0.860	0.796
1.02	0.92	0.861	0.8019

Table 4.1: Best fitness value for best value of inertia weight ω and $\propto$ (Average of 10 runs)

Inertia weight (ω)	$\propto$=1	$\propto$=0.85	$\propto$=0.7
1.01	16.8	27	29.1
1.02	18.3	29	30.9

Table 4.2: Highest numbers of times global best found for best values of inertia weight ω and $\propto$ (average of 10 runs for each parameter combination)

After determining the best value of inertia weight ω=1.02, we investigated the effect of different values of number of iterations (generations), K parameter of KNN and swarm size.

4.2.2 Number of iterations (generations)

Three values were tried 50, 100 and 150. Figure 4.3 shows the results. It can be seen that decreasing the number of generations to 50 decades for the average of best fitness value and increasing it to 150 did not give better best fitness value, so 100 generations were chosen.

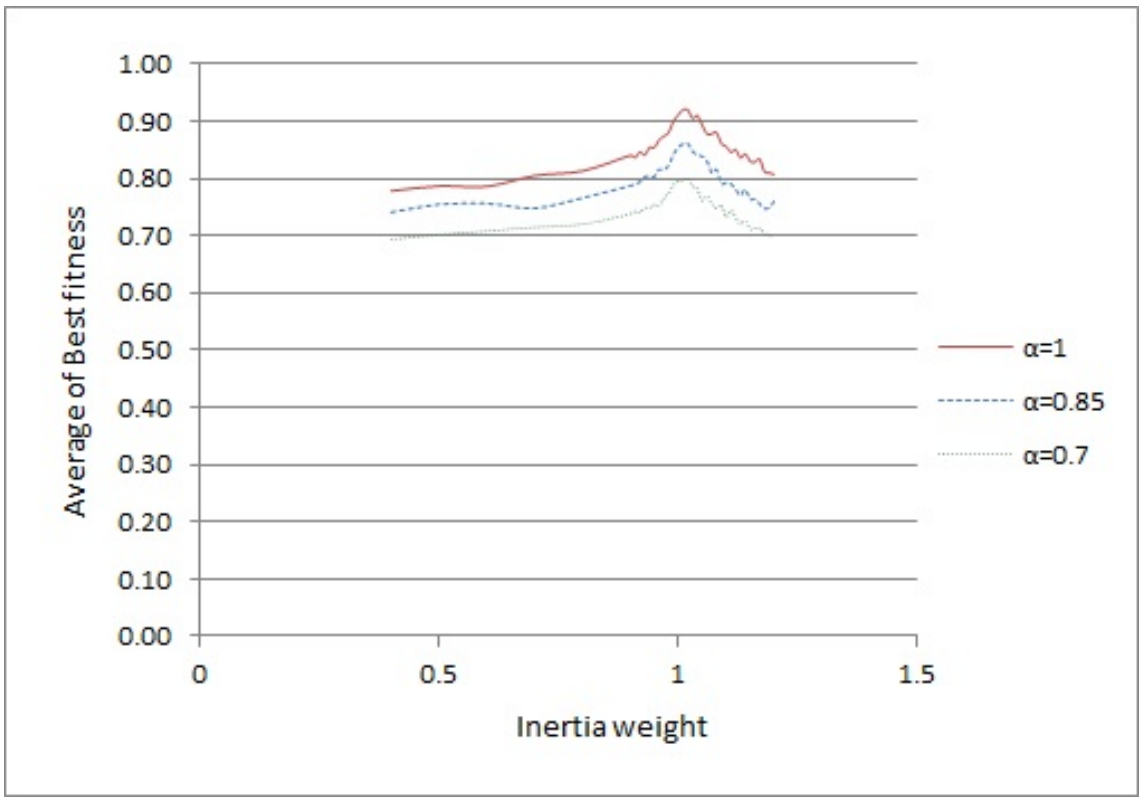

Figure 4.1: Testing different values of inertia weight (ω) to find the best value (each point is average of 10 runs)

4.2.3 K parameter for KNN classifier

Three values of K parameter were tried 1, 3 and 5. Figure 4.4 shows the obtained results. All values have given very close results with different values of $\propto$. The value of 3 was picked to be used in all experiments.

4.2.4 Swarm size

Different values of swarm size were tested 20, 30 and 40. Figure 4.5 shows the obtained results. Increasing swarm size from 30 to 40 does not yield significant improvement in terms of fitness value, so the value of 30 was selected.

4.2.5 Rare words threshold

Different values were tested to determine a suitable threshold for elimination of rare words. Experiments were done with inertia weight ω=1.02. Figure 4.6 shows the

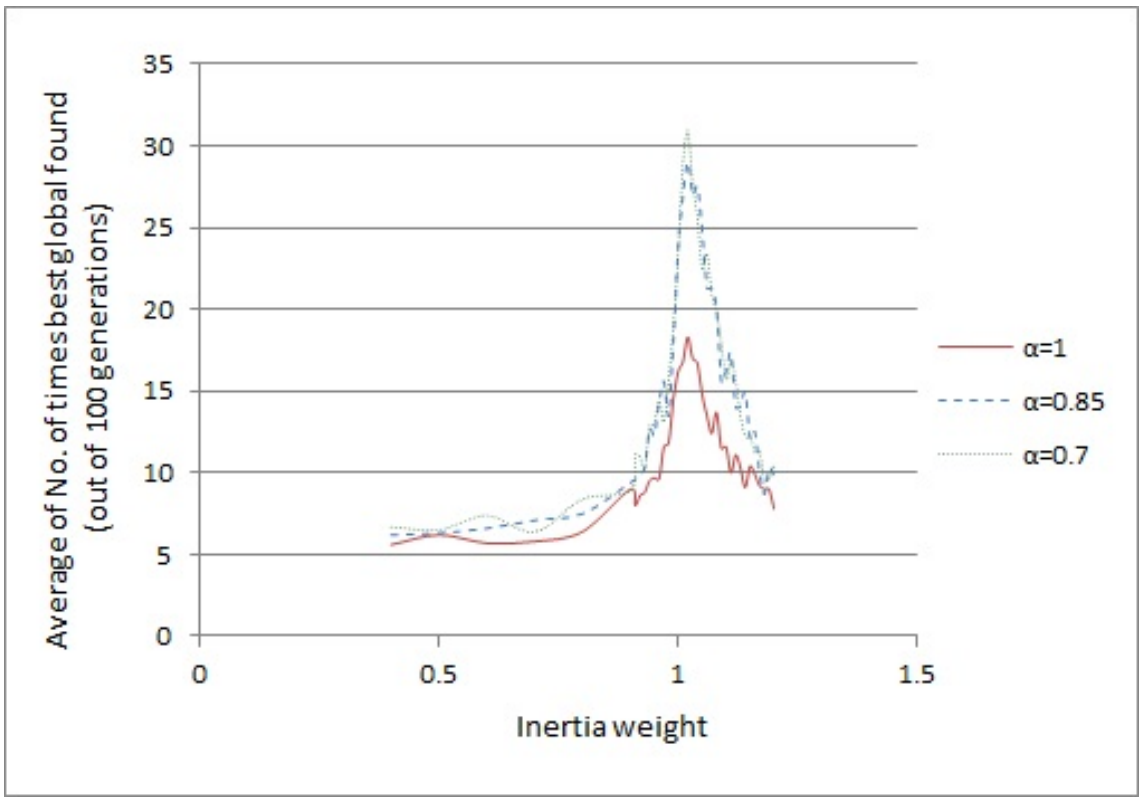

Figure 4.2: Number of times BPSO succeeded to find global best out of 100 iterations (each point is average of 10 runs)

results. It can be seen that a value of 5 is slightly better than 3 and 7 when $\propto$=1 and $\propto$=0.85, so it was selected as the best threshold for rare words removal.

Based on the obtained empirical results, the final parameters were set as the following:

- Inertia weight (ω): 1.02.
- Number of generations: 100.
- K parameter for KNN classifier: 3.
- Swarm size: 30.
- Rare words: less than 5 times were removed.
- $\propto$=0.85, β = 1 - $\propto$ =0.15.

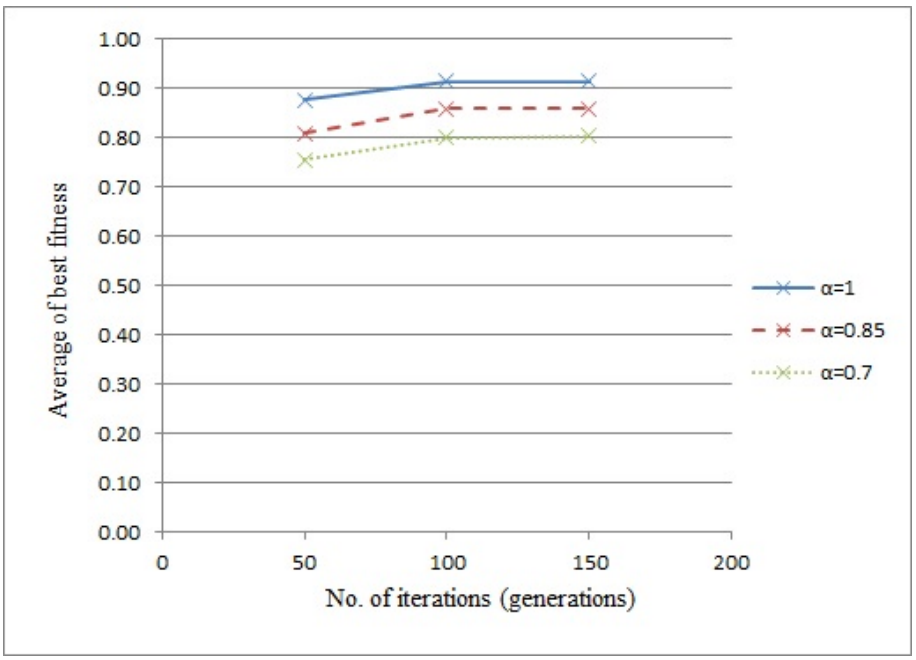

Figure 4.3: Results of using different generation for BPSO

4.3 BPSO/KNN Experiments

In this section, we will demonstrate that BPSO/KNN can be applied successfully to the Arabic classification problem. All experiments made use of the Weka open source machine learning software [50]. We selected three classifiers to evaluate the selected subsets of features for each dataset. These classifiers are:

- SVM support vector machine (with linear kernel).
- Naive Bayes classifier.
- J48 (weka implementation of C4.5).

In each case, 10-fold cross validation was used, yielding the results in the following tables. In more detail, for each of the three datasets, the following was repeated ten times:

- BPSO/KNN was run on the training set to produce a feature subset (the final *gbest* particle).
- This feature subset was used to filter the test set, i.e. the test set was processed into TFIDF vectors with components only for the given set of terms.
- SVM, Naive Bayes, and J48 were run on this test set, using ten-fold cross-validation, leading to the accuracy values that we later report.

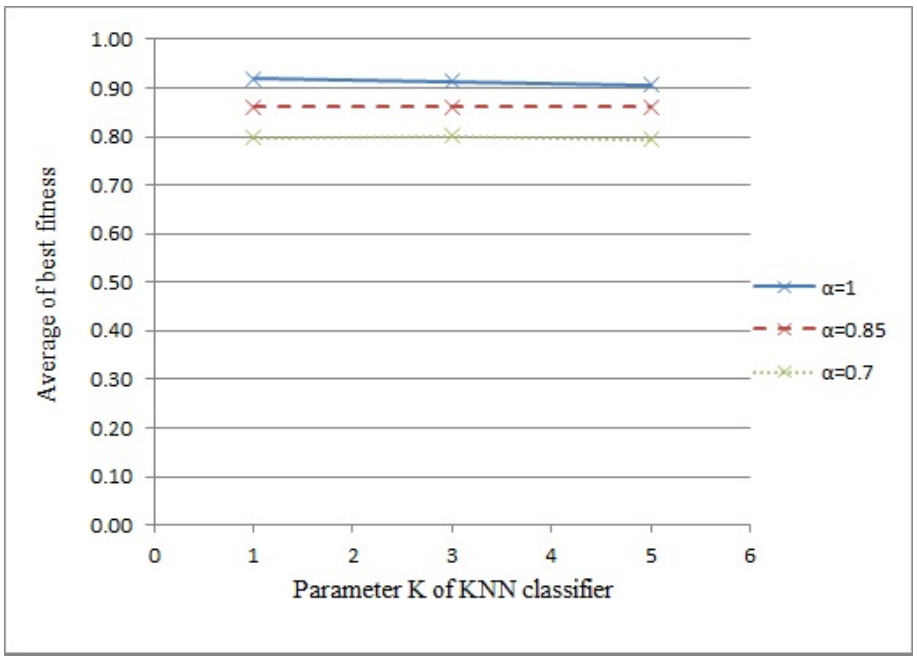

Figure 4.4: Results of using different values for K parameter of KNN

First, we show in Figures 4.7, 4.8 and 4.9 the results of ten runs on each dataset separately (each point is weighted as average F-measure). On all of the three datasets, it can be seen that SVM has the highest F-measure in all trials while C4.5 (J48) has the lowest ones. For more details, the results of SVM on *Alwatan* dataset fall in range between 0.95 and 0.964 while on *Akhbar-Alkhaleej* dataset are in range between 0.857 and 0.907. For C4.5, the results on *Alwatan* dataset are within the range 0.758 and 0.828 while on *Akhbar-Alkhaleej* dataset are between 0.759 and 0.817. For NB classifier, in all ten trials on the three datasets, it is clear that Naive Bayes is superior to C4.5 and has less performance than SVM. These signals confirm that SVM is the best among the evaluated machine learning classifiers in this work for Arabic TC task. The results also show that Naive Bayes classifier works well for this task.

In Table 4.3, we show the overall summarized results on the test set in terms of F-measure (recall this is an average of 10 complete trials of the training/test process).

For more detailed views of the results, we will show details of one of the ten trials on each of the three datasets. As in Table 4.4, we note that the sizes of the feature sets returned by BPSO (shown here rounded to the nearest unit) tended to be a little more than half of the total number of features for the dataset in question. Clearly,

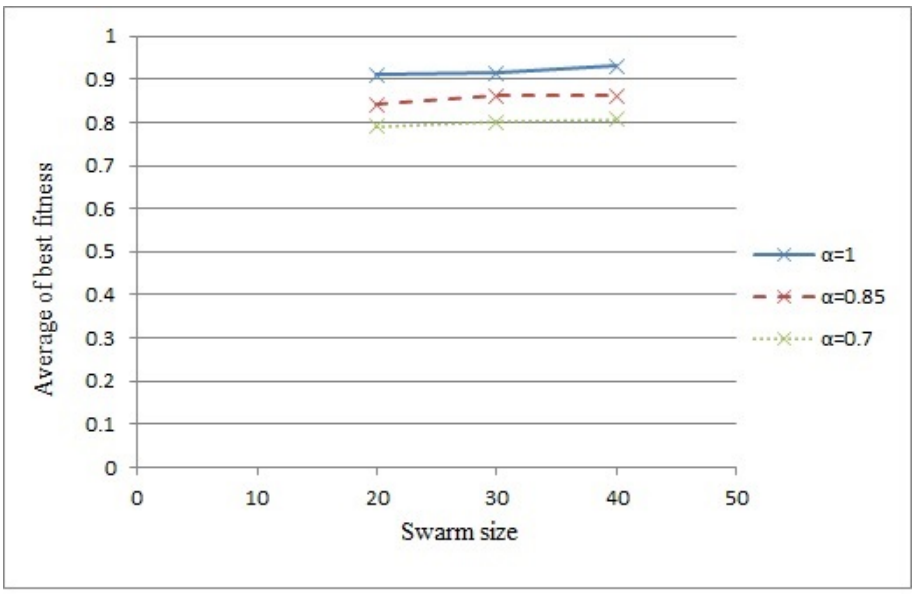

Figure 4.5: Results of using different values of swarm size

Dataset	C4.5	NB	SVM
Alj-News	**0.7396**	**0.8003**	**0.8951**
Std. Dev.	0.02211	0.02324	0.02319
Alwatan	**0.7897**	**0.8949**	**0.9597**
Std. Dev.	0.02502	0.04098	0.00424
Akhbar-Alkhaleej	**0.7853**	**0.833**	**0.8886**
Std. Dev.	0.02206	0.01036	0.01608

Table 4.3: Classification accuracy of SVM, Naive Bayes and C4.5 on the three datasets using BPSO/KNN

SVM was able to classify most accurately, with results that seem quite competitive given the results that tend to be achieved in this research area.

Tables 4.5 to 4.13 set out more detailed views of the results on each of the three datasets, showing mean values for precision, recall and F-measure for each category in the dataset in question, where the averages are weighted according to the numbers of documents in each category. Tables 4.5, 4.6, and 4.7 respectively show the results for SVM, Naive Bayes, and J48 on the *Alj-News* dataset, while the sequence is repeated for *Akhbar-Alkhaleej* dataset in Tables 4.8 to 4.10 and *Alwatan* dataset in Tables 4.11 to 4.13.

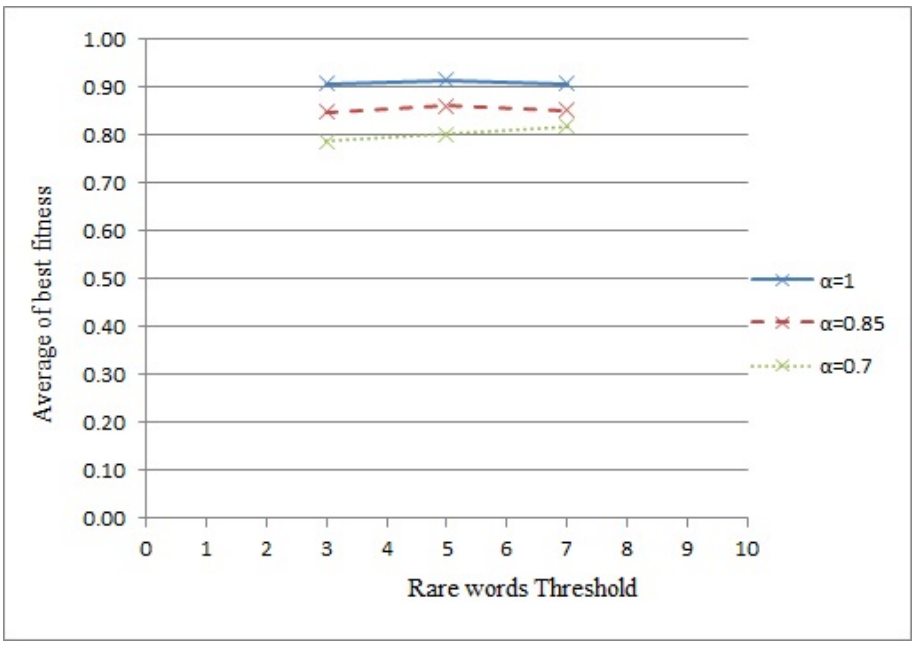

Figure 4.6: Results of using different threshold for rare words elimination

Dataset	*Distinct features from the training set*	*Selected features by BPSO/KNN*
Alj-News	5329	**2841**
Alwatan	12282	**6894**
Akhbar-Alkhaleej	8913	**4638**

Table 4.4: Selected features by BPSO/KNN for a specific trial out of ten

For *Alj-News* dataset, as shown in Tables 4.5, 4.6, and 4.7 respectively, it is clear that SVM outperformed both NB and C4.5. As shown in Table 4.5, the precision and recall of the Sport and Science categories were higher than other categories. The Politics category has the lowest precision. It seems that SVM incorrectly classified some documents from Art and Economic categories by assigning them to the Political category. In addition, the precision of the Sport category is 1 and the F-measure is 0.983; this indicates that SVM correctly separated other classes from the Sport category by not assigning any documents from the other three categories to the Sport category. Although SVM wrongly assigned a few sport documents to other categories.

From Table 4.6, it can be seen that NB worked well from the *Alj-News Dataset*. The lowest precision was 0.654 by the Politics category and the lowest recall was

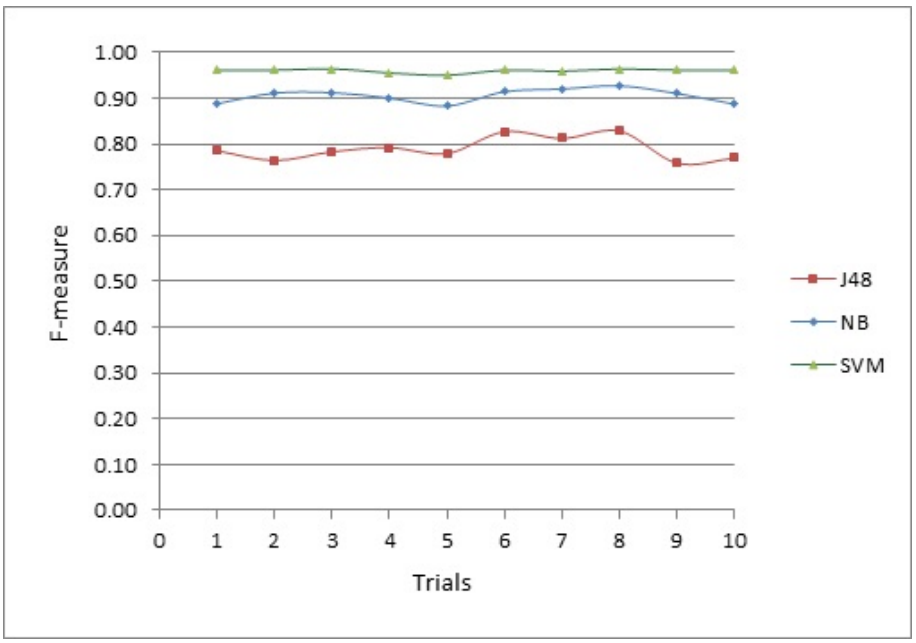

Figure 4.7: Results of ten runs on Alwatan dataset using BPSO/KNN

by the Art category. That means SVM incorrectly classified some instances to not be related to the Art category while in fact they are. It also incorrectly assigned some documents actually related to the Politics category to other classes. As can be noticed from Table 4.7, a similar difficulty faced C4.5 to properly distinguish between the Politics and Art categories. In comparison with SVM and NB, C4.5 did more mistakes in this context.

Class	**Precision**	**Recall**	**F-Measure**
Sport	1	0.967	0.983
Art	0.898	0.883	0.891
Science	0.982	0.933	0.957
Politics	0.781	0.95	0.857
Economic	0.925	0.817	0.867
Weighted Avg.	0.917	0.91	**0.911**

Table 4.5: Detailed Accuracy by Class for SVM Classifier on Alj-News Dataset

Tables 4.8, 4.9 and 4.10 show the performance of SVM, NB and C4.5 respectively on *Akhbar-Alkhaleej* dataset. From Table 4.8, it can be seen that the highest precision is for the Sport category which is 0.986. The second highest precision is for Int. news categories. The precisions of Economy and Local news are close to each

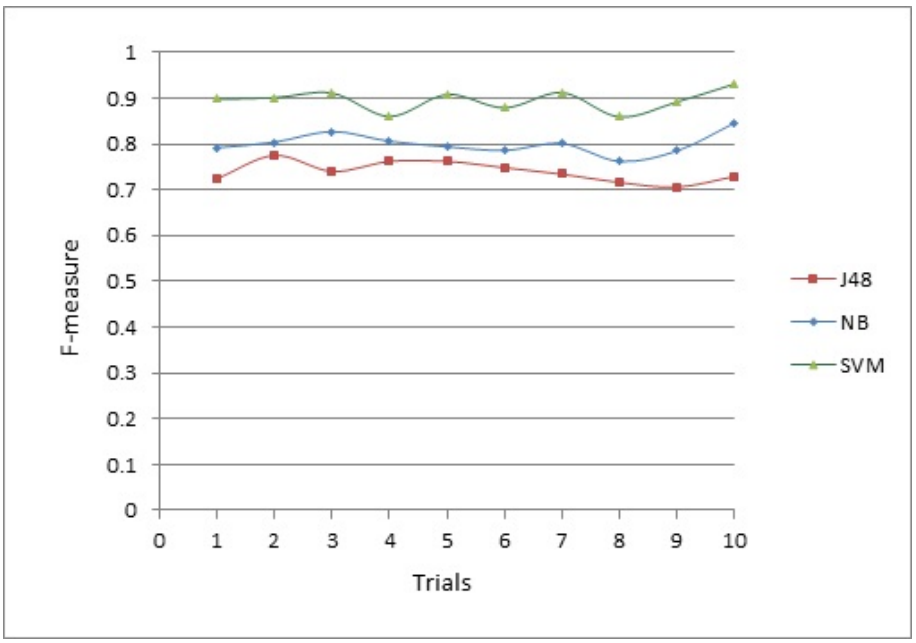

Figure 4.8: Results of ten runs on Alj-News dataset using BPSO/KNN

Class	Precision	Recall	F-Measure
Sport	0.964	0.883	0.922
Art	0.815	0.733	0.772
Science	0.925	0.817	0.867
Politics	0.654	0.85	0.739
Economic	0.833	0.833	0.833
Weighted Avg.	0.838	0.823	**0.827**

Table 4.6: Detailed Accuracy by Class for Naive Bayes Classifier on Alj-News Dataset

other and both are less than 0.9. This could be interpreted that SVM wrongly classified some documents from Sport and Int. News categories to Economy and Local news categories. For NB classifier, as in Table 4.9, it is clear from the precision of Economy category that NB incorrectly classified documents from other categories to Economy class. The recall of Local news category indicates that the wrongly assigned documents to economy category could be from Local news category. In contrast with SVM and NB classifiers, Table 4.10 shows that C4.5 has the lowest performance on *Akhbar-Alkhaleej* dataset.

Tables 4.11, 4.12 and 4.13 show the performance of SVM, NB and C4.5 respectively on *Alwatan* dataset. In general, SVM is best for predicting the correct

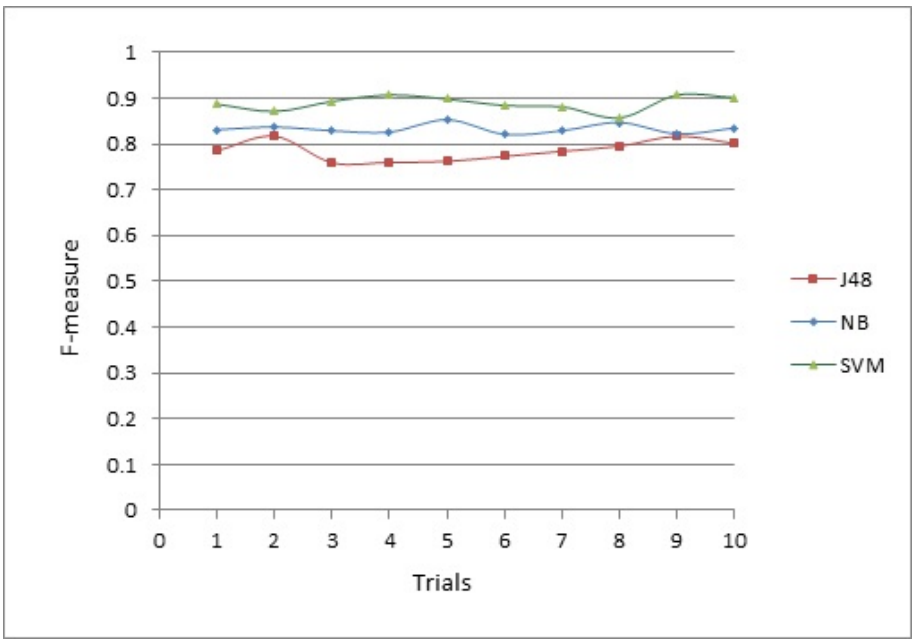

Figure 4.9: Results of ten runs on Akhbar-Alkhaleej dataset using BPSO/KNN

Class	Precision	Recall	F-Measure
Sport	0.869	0.883	0.876
Art	0.711	0.533	0.61
Science	0.909	0.833	0.87
Politics	0.562	0.683	0.617
Economic	0.697	0.767	0.73
Weighted Avg.	0.75	0.74	**0.74**

Table 4.7: Detailed Accuracy by Class for C4.5 Classifier on Alj-News Dataset

categories of given test documents. As shown in Table 4.11, the precision of the Religion and Sport categories indicates that SVM did not assign any document related to Economy or Culture categories to the Religion or Sport categories. For NB classifier, it can be seen from Table 4.12 that the precision of Culture category is the lowest. This means that NB incorrectly classified documents from other categories to the culture category. From Table 4.13, the precision of the Culture and Economy categories shows that C4.5 could not differentiate between economy and culture documents.

Tables from 4.14 to 4.22 show the confusion metrics. The confusion between topics is generally understandable given the levels of similarity between the topics in

Class	Precision	Recall	F-Measure
Economy	0.836	0.836	0.836
Int. News	0.943	0.862	0.901
Local News	0.846	0.917	0.88
Sport	0.987	0.907	0.945
Weighted Avg.	0.896	0.892	**0.893**

Table 4.8: Detailed Accuracy by Class for SVM Classifier on Akhbar-Alkhaleej Dataset

Class	Precision	Recall	F-Measure
Economy	0.671	0.927	0.779
Int. News	0.929	0.897	0.912
Local News	0.84	0.764	0.8
Sport	0.888	0.826	0.855
Weighted Avg.	0.84	0.828	**0.829**

Table 4.9: Detailed Accuracy by Class for Naive Bayes Classifier on Akhbar-Alkhaleej Dataset

different cases. For example, the dataset that is associated with the lowest accuracy values, *Akhbar-Alkhaleej*, needs the classifier to distinguish between international news, local news, economy and sport. Since either local news or international news can often be about sport and/or the economy, it is not surprising that any automated method could be quite challenged to predict the labelled categories. Similar potential for confusion exists in each dataset, but to a lesser extent, and we see each of these observations reflected in the confusion matrices, as well as the overall results.

Tables 4.14, 4.15, and 4.16 present the confusion metrics of SVM, NB and C4.5 for *Alj-News* dataset. As shown in Table 4.14, for the economic category, SVM incorrectly classified eleven test documents out of sixty. Eight documents belonging to economic class were classified as politics. A similar case happened with the Art category, six documents were wrongly assigned to the Politics category. Similar mistakes were done by NB. It can be seen from Table 4.15 that NB misclassified sixteen documents originally related to the Art category by assigning nine of them to politics class, two documents to science class, two documents to sport class and

Class	Precision	Recall	F-Measure
Economy	0.653	0.582	0.615
Int. News	0.754	0.741	0.748
Local News	0.757	0.778	0.767
Sport	0.831	0.86	0.846
Weighted Avg.	0.758	0.761	**0.759**

Table 4.10: Detailed Accuracy by Class for C4.5 Classifier on Akhbar-Alkhaleej Dataset

Class	Precision	Recall	F-Measure
Culture	0.853	0.955	0.901
Economy	0.939	0.928	0.933
Religion	1	0.978	0.989
Sport	1	0.954	0.977
Weighted Avg.	0.958	0.955	**0.955**

Table 4.11: Detailed Accuracy by Class for SVM Classifier on Alwatan Dataset

three documents to economic class. In the case of C4.5 as presented in Table 4.16, 28 documents out of sixty which actually belong to the Art category were incorrectly assigned to other categories.

For *Akhbar-Alkhaleej* dataset, from Table 4.17, it can be seen that SVM misclassified eight documents related to Local news by classifying them as economy documents. Also, nine documents actually related to economy were classified as local news. In the case of NB classifier, as shown in Table 4.18, in comparison with SVM, NB classified more documents originally related to local news to economy class. In addition, both SVM and NB assigned some documents from the Sport category as Local news. For C4.5, Table 4.19 has revealed that Sport, International and Economy categories overlapped with local news.

In the case of *Alwatn* dataset, confusion matrices in Tables 4.20, 4.21 and 4.22 show that most misclassification cases were caused by the Culture category. In Table 4.20, six documents belonging to the Economy category were assigned to Culture. In Table 4.21, six, eight and seven documents relating to Economy, Religion and

Class	Precision	Recall	F-Measure
Culture	0.72	0.806	0.761
Economy	0.928	0.928	0.928
Religion	0.914	0.914	0.914
Sport	0.99	0.917	0.952
Weighted Avg.	0.904	0.898	**0.90**

Table 4.12: Detailed Accuracy by Class for Naive Bayes Classifier on Alwatan Dataset

Class	Precision	Recall	F-Measure
Culture	0.642	0.507	0.567
Economy	0.674	0.771	0.719
Religion	0.933	0.903	0.918
Sport	0.86	0.899	0.879
Weighted Avg.	0.794	0.795	**0.792**

Table 4.13: Detailed Accuracy by Class for C4.5 Classifier on Alwatan Dataset

Sport respectively were wrongly classified by NB as Culture. It can be seen in Table 4.22 that C4.5 incorrectly classified 33 documents originally belonging to Culture category to other categories. 22 texts out of 33 were assigned to the Economy category. This reflects the fact that it is a challenge for C4.5 to distinguish between naturally close categories such as Culture and Economy.

It is clear that SVM is the best one among the tested classifiers for the Arabic text classification task. Also, Naive Bayes classifier works well for this task. In contract with other applied classifiers for Arabic TC task, C4.5 has recorded the lowest accuracies in all presented results.

4.4 Effect of using Normalization and light stemming on BPSO/KNN performance

This section investigates the effect of normalization and light stemming on the classification performance of SVM, NB and C4.5 using BPSO/KNN as a feature selec-

a	b	c	d	e	⟸ classified as
58	1	0	1	0	a = sport
0	**53**	0	6	1	b = art
0	2	**56**	1	1	c = science
0	1	0	**57**	2	d = politics
0	2	1	8	**49**	e = economic

Table 4.14: Confusion Matrix (SVM on Alj-News Dataset)

a	b	c	d	e	⟸ classified as
53	2	1	4	0	a = sport
2	**44**	2	9	3	b = art
0	2	**49**	5	4	c = science
0	5	1	**51**	3	d = politics
0	1	0	9	**50**	e = economic

Table 4.15: Confusion Matrix (Naive Bayes on Alj-News Dataset)

tion method. All experiments were repeated after adding normalization and light stemming operations to the text pre-processing steps. Normalization is performed according to the following steps [29,31, 32]:

- Replace Alef "أ ، إ " and " آ " with "ا".
- Replace Yeh "ي" with "ى".
- Replace " ة " with "ه".
- Remove Arabic diacritics and stretching character (Tatweel).

As mentioned earlier, Arabic Light stemmer (light10) does not affect the meaning of words. It only removes the conjunction "Wa" " "و" , some prefixes like:

(ف ، ك ، ل ، ب ، ال ، وال ، بال ، كال ، فال ، لل)

and some suffixes such as:

(ها ، ون ، وا ، ين ، ان ، يه ، ية ، ا ، ة ، ه) [31].

Using Java libraries for Arabic normalization and light stemming integrated with the Apache Lucene software (those libraries are available at [31, 32]), the numbers of distinct features found in the separate datasets were as shown in Table 4.23. It

a	b	c	d	e	⟸ classified as
53	1	1	4	1	a = sport
5	**32**	2	13	8	b = art
1	2	**50**	5	2	c = science
1	7	2	**41**	9	d = politics
1	3	0	10	**46**	e = economic

Table 4.16: Confusion Matrix (C4.5 on Alj-News Dataset)

a	b	c	d	⟸ classified as
46	0	9	0	a = Economy
0	**50**	8	0	b = International
8	3	**132**	1	c = Local
1	0	7	**78**	d = Sport

Table 4.17: Confusion Matrix (SVM on Akhbar-Alkhaleej Dataset)

is clear that normalization and light stemming have led to a significant reduction in the number of distinct features.

Table 4.24 presents the obtained results after adding normalization and light stemming to the pre-processing. In general, normalization and light stemming have led to a great reduction in the number of distinct features and also has resulted in statistically significant improvement of the classification accuracy of NB classifier on the three applied datasets.

Figure 4.10 presents the difference between with and without applying normalization and light stemming in the pre-processing stage. It seems that normalization and light stemming have improved the performance of SVM, NB and C4.5 slightly. However, in most cases the difference is small. Standard T-test (one-tailed) with significant level by $p < 0.1$ was applied to measure the significance of the improvement. On *Alwatan* dataset, as shown in Figure 4.10, the SVM results with normalization and light stemming are superior to those without normalization and light stemming ($p < 0.005$). Also, the results of C4.5 with normalization and light stemming outperformed those without normalization and light stemming ($p < 0.002$) while the results of NB are not significant.

a	b	c	d	⟸ classified as
51	0	3	1	a = Economy
2	**52**	4	0	b = International
22	4	**110**	8	c = Local
1	0	14	**71**	d = Sport

Table 4.18: Confusion Matrix (Naive Bayes on Akhbar-Alkhaleej Dataset)

a	b	c	d	⟸ classified as
32	5	17	1	a = Economy
4	**43**	8	3	b = International
13	8	**112**	11	c = Local
0	1	11	**74**	d = Sport

Table 4.19: Confusion Matrix (C4.5 on Akhbar-Alkhaleej Dataset)

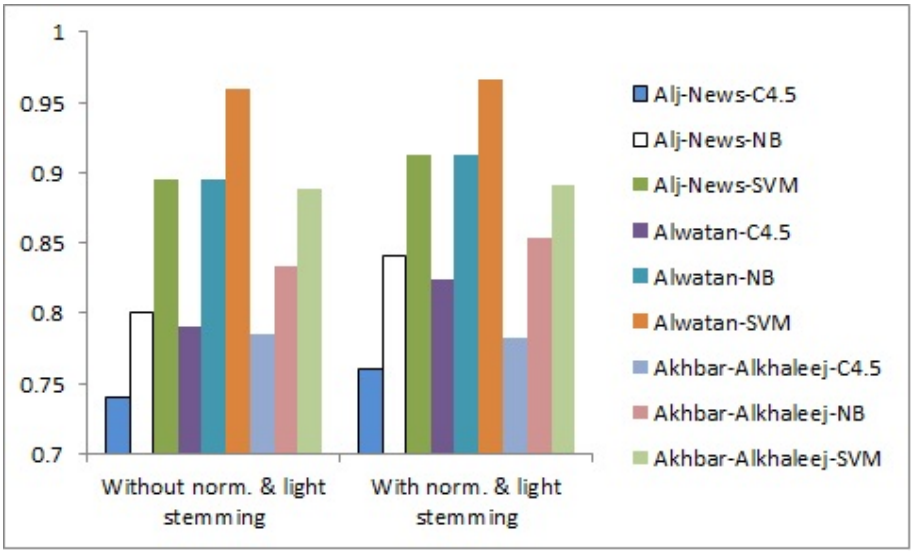

Figure 4.10: Classification accuracy of SVM, NB and C4.5 with and without normalization and light stemming (BPSO/KNN)

Figure 4.10 shows the results on Alj-News dataset, generally, normalization and light stemming has resulted in a better performance for the three applied algorithms. The results for SVM with normalization and light stemming are superior to those without normalization and light stemming ($p < 0.03$). In addition, the results of NB with normalization and light stemming outperformed those without normalization and light stemming ($p < 0.0003$). Also, the results for C4.5 with normalization

a	b	c	d	⟸ classified as
64	3	0	0	a = Culture
6	**77**	0	0	b = Economy
2	0	**91**	0	c = Religion
3	2	0	**104**	d = Sport

Table 4.20: Confusion Matrix (SVM on Alwatan Dataset)

a	b	c	d	⟸ classified as
54	4	8	1	a = Culture
6	**77**	0	0	b = Economy
8	0	**85**	0	c = Religion
7	2	0	**100**	d = Sport

Table 4.21: Confusion Matrix (Naive Bayes on Alwatan Dataset)

and light stemming are better than those without normalization and light stemming ($p < 0.015$). On *Akhbar-Alkhaleej* dataset, as can be seen in Figure 4.10, for SVM and C4.5, it seems that the results with applying normalization and light stemming are better than no normalization and light stemming. Standard T-test (one-tailed) with significant level by $p < 0.1$ has shown no significant differences, however the corresponding results of NB with normalization and light stemming are superior to those without normalization and light stemming ($p < 0.008$).

Standard T-test (one-tailed) with significant level by $p < 0.1$ (confidence level 90%) was applied to see which method is better (with or without normalization and light stemming using BPSO/KNN for feature selection). Considering the full sets of results using BPSO/KNN, on Alwatan dataset, a one tailed-T-test shows that BPSO/KNN with normalization and light stemming is significantly better than BPSO/KNN without normalization and light stemming ($p < 0.005$) with confidence level 99%. Also, on Alj-News datasets, BPSO/KNN with normalization and light stemming is significantly better than BPSO/KNN without normalization and light stemming ($p < 0.005$) with confidence level 99%. On Akhbar-Alkhaleej dataset, 99% confidence that BPSO/KNN with normalization and light stemming is better ($p < 0.007$).

a	b	c	d	⟸ classified as
34	22	3	8	a = Culture
8	**64**	3	8	b = Economy
5	4	**84**	0	c = Religion
6	5	0	**98**	d = Sport

Table 4.22: Confusion Matrix (C4.5 on Alwatan Dataset)

Dataset	Distinct features from the training set	
	Without normalization and light stemming	With normalization and light stemming
Alj-News	5329	3763
Alwatan	12282	7062
Akhbar-Alkhaleej	8913	5367

Table 4.23: Distinct features from the three datasets with and without normalization and light stemming

4.5 Summary

In this chapter, BPSO-KNN was proposed as a feature selection method for Arabic text classification. Three Arabic datasets were used to test this method, and three well-known machine learning algorithms SVM , Naive Bayes and C4.5 decision tree learning (in its Weka implementation as J48) were applied to classify Arabic test documents using features selected by this method.

The first set of experiments in this chapter was performed to find out appropriate values for the parameters of the BPSO algorithm. Those parameters include: inertia weight, number of generations, swarm size and K parameter for KNN classifier. Then, on each training set of the three Arabic datasets described in previous chapter, BPSO/KNN was trialed ten times. The selected subsets features, in conjunction with SVM, Naive Bayes and C4.5 classifiers were used to classify the hold out test sets.

Our results suggest that the proposed method is effective. It led to values for classification accuracy and F1-measure that compare well with those reported in

Dataset	C4.5	NB	SVM
Alj-News	**0.7606**	**0.8406**	**0.9121**
Std. Dev.	0.01663	0.01796	0.01187
P-value	0.01417	0.00023	0.0295
Alwatan	**0.8237**	**0.9126**	**0.9667**
Std. Dev.	0.02023	0.01009	0.00618
P-value	0.00191	0.10699	0.0047
Akhbar-Alkhaleej	**0.7825**	**0.8529**	**0.8914**
Std. Dev.	0.01827	0.02027	0.01663
P-value	0.38042	0.00786	0.35323

Table 4.24: Classification accuracy of SVM, NB and C4.5 on the three datasets with normalization and light stemming (BPSO/KNN)

related work. However a direct comparison with related work in this area is not currently possible, since either the datasets used in an associated publication are not available, or, where they are available, we have been unable to discover the way the data was split into training and/or validation and/or test data in the comparative results. Therefore another contribution of this work is to make our datasets available, with the latter issues clarified, to support continuing work on this topic. Meanwhile, it seems clear that the results achieved by SVM, as well as Naive Bayes, overall, suggest that BPSO/KNN performs well as a feature selection technique for this task (well enough, for example, to underpin the selection of features for associated applications).

The effect of normalizing some Arabic letters and applying Arabic light stemming on the performance of SVM, NB and C4.5 with BPSO/KNN feature selection method was investigated. The results were compared against the results when normalization and light stemming were not applied. In almost all experiments using normalization and light stemming, the averages of results (F-measure) for the three ML classifiers on the three Arabic datasets are superior to the results when normalization and light stemming were not used.

Chapter 5

Feature Subset selection for Arabic TC using BPSO with SVM

In the previous chapter, we explored binary particle swarm optimization hybridized with K nearest neighbour for feature selection in Arabic document categorization task. The experimental results have shown that this feature selection method works well for selecting good sets of features for the Arabic TC problem. In this chapter, we demonstrate a combination of Binary PSO and Support Vector Machine that performs well in selecting good sets of features for the Arabic document categorization task. Also, we compare its performance with our previously presented BPSO/KNN feature selection approach. Moreover, we also analyze the set of features and consider the differences between the sort of features that BPSO/KNN and BPSO/SVM tend to select.

5.1 BPSO/SVM feature selection method

The difference between BPSO/KNN and BPSO/SVM is that in the fitness function. The classification accuracy (Acc) of a particle (P) on the training set is estimated using the SVM classifier instead of KNN.

The classification accuracy of a particle (P) is calculated using the following procedure:

- Filter the subset of features selected by P in the training set.
- Evaluate the filtered subset using the SVM classifier (using Weka SVM Library

developed by [117]) with 10 fold cross validation.

- Use the accuracy (F-measure) of SVM to calculate the fitness of a particle.

5.2 BPSO/SVM Experiments

In this section, we will demonstrate that BPSO/SVM can be applied successfully to the Arabic classification problem. The same classifiers used with BPSO/KNN were applied to evaluate the proposed method.

Also, the same parameters for BPSO were used:

- Inertia weight (ω): 1.02.
- Number of generations: 100.
- Swarm size: 30.
- Rare words: less than 5 times were removed.
- $\propto$=0.85, β = 1 - $\propto$ =0.15.

SVM parameters were set as follows:

- Learning algorithm: C-SVC (Support Vector Classification).
- Linear kernel function.
- The cost parameter C=1 (a relatively low value favouring good generalization).

Similar to BPSO/KNN, in each case, 10-fold cross validation was used, yielding the results in the following tables. In more detail, for each of the three datasets, the following was repeated ten times:

- BPSO/SVM was run on the training set to produce a feature subset (the final *gbest* particle).
- This feature subset was used to filter the test set, i.e. the test set was processed into TFIDF vectors with components only for the given set of terms.
- We used three different classification algorithms to evaluate the selected subsets of features on a holdout/test portion of each dataset. SVM, NB, and C4.5 were all run on the test set.

Figures 5.1, 5.2 and 5.3 present the results of ten runs on each dataset separately (each points is weighted average F-measure of the training/test process). On all of the three datasets, it can be seen that SVM has the highest F-measure in all trails while C4.5 (J48) has the lowest ones. In detail, the results of SVM on the *Alwatan* dataset fall between the range 0.953 and 0.977 while on the *Akhbar-Alkhaleej* dataset they are within the range between 0.872 and 0.908. For C4.5, the results on the *Alwatan* dataset are between 0.758 and 0.828 whereas on the *Akhbar-Alkhaleej* dataset they are between 0.765 and 0.833. For NB classifier, in all ten trials on the three datasets, it is clear that Naive Bayes is superior to C4.5 and has worse performance than SVM. Using feature subsets selected by the BPSO/SVM FS method, these observations confirm that SVM is the best among the evaluated machine learning classifiers in this work for the Arabic TC task. The results also indicate that the Naive Bayes classifier has performed well for this task.

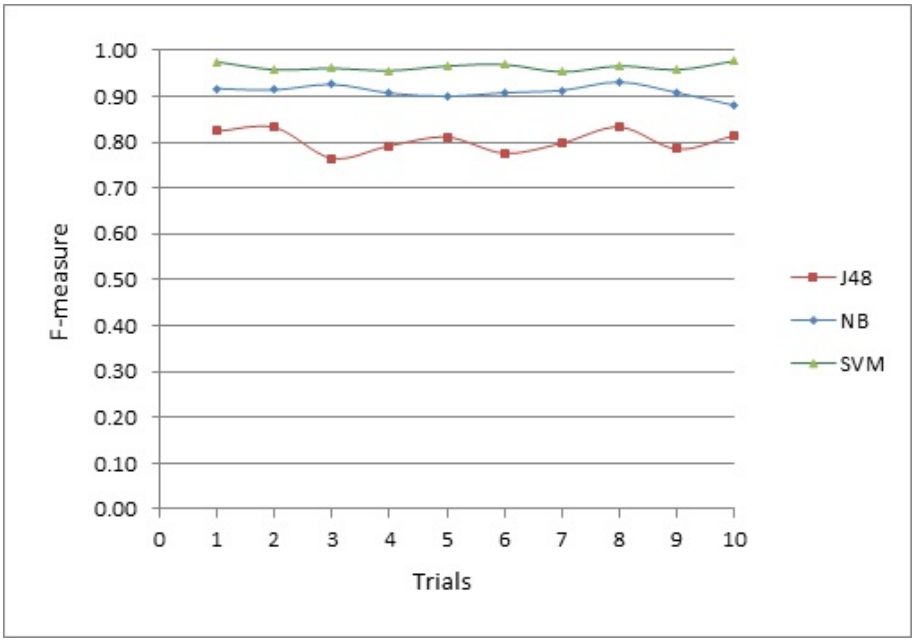

Figure 5.1: Results of ten runs on *Alwatan* dataset (BPSO/SVM)

In Table 5.1, we show the overall summarised results on the test set (using BPSO/SVM as a feature selection method) in terms of F-measure (recall this is an average of 10 complete trials of the training/test process).

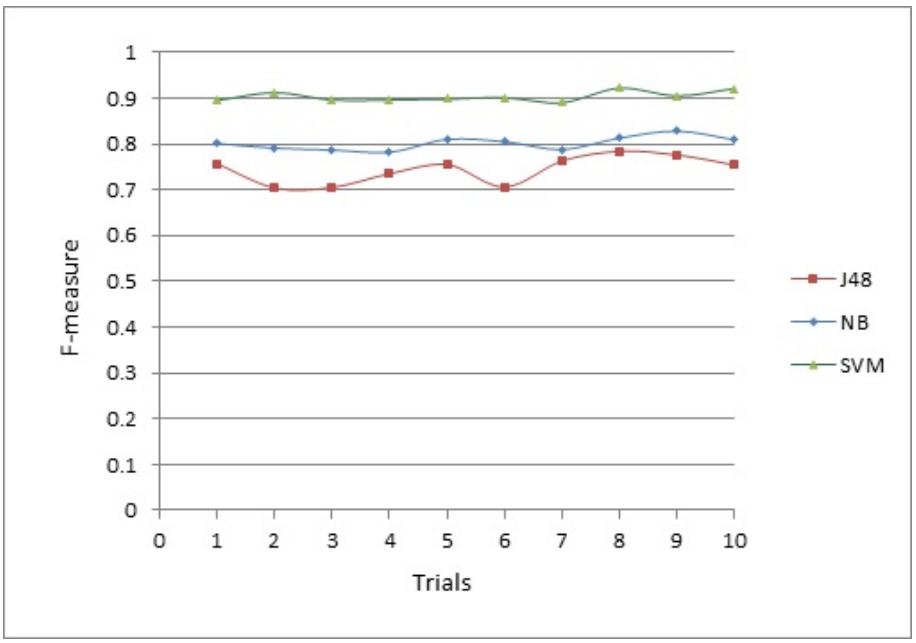

Figure 5.2: Results of ten runs on *Alj-News* dataset (BPSO/SVM)

5.3 Comparison between BPSO with KNN and BPSO with SVM

BPSO/SVM and BPSO/KNN were each trialled ten times independently on each of the three datasets. After each independent trial, the best resulting feature subset was applied to reduce the hold set portion of the dataset, and C4.5, NB and SVM were independently applied, using tenfold cross validation. As can be seen from Figure 5.4, basic inspection finds that for each dataset and learning method, the mean accuracy on the holdout set is higher for BPSO/SVM than for BPSO/KNN. However, in most cases the difference is small, but we think this is because we are getting close to the maximum accuracy achievable. Standard T-test was applied (one-tailed), with significant signalling by $p < 0.1$, and found the following. On *Alwatan*, the SVM results for BPSO/SVM are superior to those of BPSO/KNN ($p < 0.085$); however, the corresponding results for NB and C4.5 are not significant. On *Alj-News*, none of the corresponding pair wise comparisons are significant, while on *Akhbar-Alkhaleej*, the NB results for BPSO/SVM are superior to those of BPSO/KNN ($p < 0.044$), while the other two comparisons show no significant difference.

Standard T-test (one-tailed) with significant level by $p < 0.1$ (confidence level

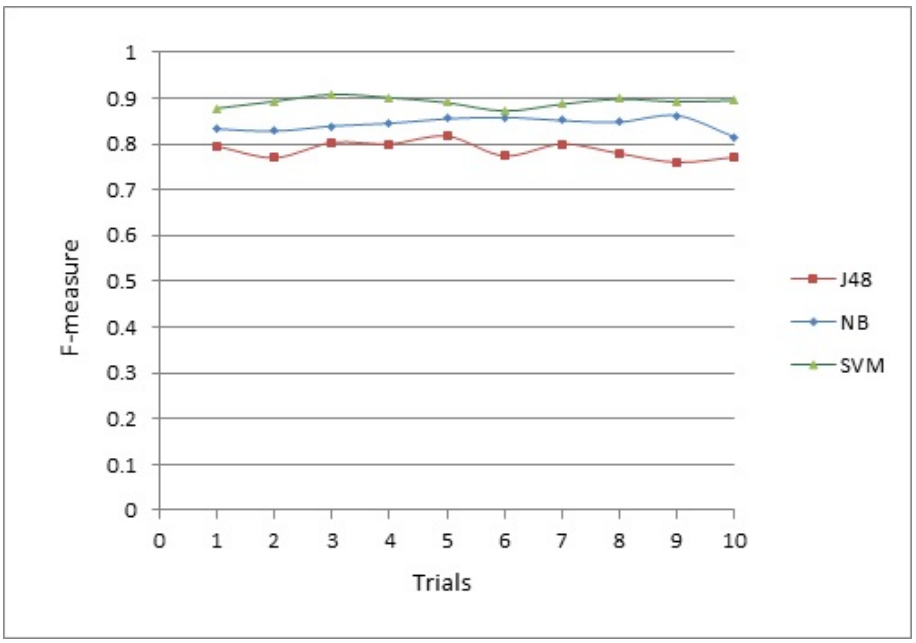

Figure 5.3: Results of ten runs on*Akhbar-Alkhaleej* dataset (BPSO/SVM)

90%) was applied to see which method is better (BPSO/KNN or BPSO/SVM) selection). Considering the full sets of results using BPSO/KNN and BPSO/SVM, on Alwatan dataset, one tailed-T-test shows that BPOS/SVM is significantly better than BPSO/KNN ($p < 0.06$) with confidence level 94%. On Alj-News and Akhbar-Alkhaleej datasets, BPSO/SVM has better averages than BPSO/KNN, but one-tailed T-tests reveal that these differences are not significant, and we cannot be sure which of BPSO/SVM or BPSO/KNN is the best method.

We performed further analysis in an attempt to understand the difference in performance. The selected features were recorded for every trial in each experiment. This enabled us to examine the degree to which individual features repeatedly emerged in different trial runs on the same data. We summarize the broad findings from this analysis in Table 5.2.

For example, on *Alwatan* dataset, Table 5.2 tells us there were 686 features (words) that each occurred in precisely 8 of the 10 trials of BPSO/SVM, while the corresponding number for BPSO/KNN was 1072. We also see that, for BPSO/SVM, there were 24 features present in the result of every trial, while the corresponding figure for BPSO/KNN was 38. Overall, feature sets from different trial runs of BPSO/KNN tend to have a greater overlap with each other than those emerging

Dataset	J48	NB	SVM
Alj-News	**0.7439**	**0.8014**	**0.9036**
Std. Dev.	0.02985	0.01437	0.01119
P-value	0.35949	0.4502	0.1578
Alwatan	**0.8031**	**0.9104**	**0.9639**
Std. Dev.	0.02365	0.01409	0.00809
P-value	0.11716	0.14092	0.08432
Akhbar-Alkhaleej	**0.7869**	**0.8434**	**0.8914**
Std. Dev.	0.01845	0.01471	0.01056
P-value	0.43118	0.04301	0.3259

Table 5.1: Classification accuracy of SVM, Naive Bayes and Decision trees on the three datasets using BPSO/SVM

from BPSO/SVM trials.

We also report that the intersection between the frequently appearing BPSO/KNN and BPSO/SVM features were always zero or negligible, e.g. on Alj-News dataset, as in tables 5.3 and 5.4 there was no overlap between the BPSO/SVM and BPSO/KNN in all 10 trials features.

The fact that each method seems to rely on a small core of features possibly reflects the semantics of document categories. That is for a category such as religion, you may expect a set of words (features) which usually indicates a strong signal about the category, while a larger number of additional words may provide a weaker signal, since they could also be common to other categories (e.g. culture or news).

However, the fact that BPSO/SVM and BPSO/KNN tend to choose distinct core sets from each other is perhaps more interesting. We expect this reflects the distinct approaches used by SVM and KNN. Features most often chosen by KNN could be core to a category in the sense that they are close to central in a document feature space. Meanwhile, the important features for SVM are those that help define clear boundaries between categories in the transformed feature space. The SVM will be more tolerant with those that are shared between categories. E.g. consider a

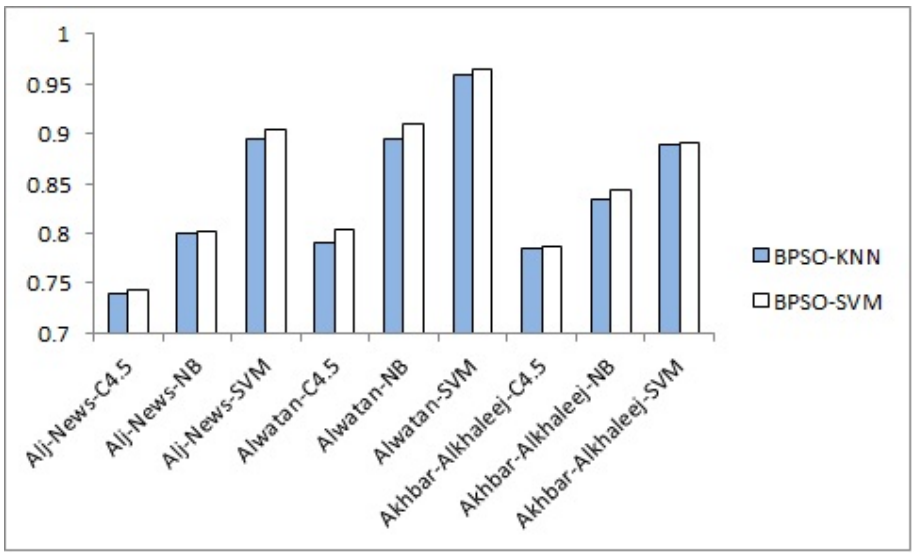

Figure 5.4: Performance of BPSO-SVM and BPSO-KNN

document in local news that is close to the boundary with international news. KNN will misclassify this document if its closest neighbours tend to be over the boundary, however SVM will classify it correctly. It can be suggested that SVM features may be chosen according to how well boundaries can be defined, and KNN features may be more biased towards a large distance between documents in different categories.

This would suggest that BPSO/SVM may be preferable when distance between categories is small, since several documents will be close to boundaries. There are hints that this reflected in our results where clear statistical significance was found among individual comparisons; this was in the case of *Alwatan* dataset, where the categories include Culture and Religion, and the *Akhbar-Alkhaleej* dataset, where the categories include International News and Local News. Figure 5.5 represents an example of what are possibly selected features by BPSO/KNN and BPOS/SVM.

Note that, in comparison with English, the definite article *the* is written differently in Arabic language. The definite Arabic noun is formed by joining the definite Article (ال) (*AL*) to the beginning of the noun. For instance, the word (الثقافة) means *the culture* and (ثقافة) means *Culture*. In this case, each word is considered as a unique token (feature). Also, the conjunction (*and*) which is (*Wa*) (و) in Arabic (consisting of one letter only) is frequently attached to the beginning of words. For example, the word (وأجهزة) which means *and devices* is considered as a different

Dataset	No. of Trials	BPSO/KNN	BPSO/SVM
Alj-News	6	1260	1112
	7	804	683
	8	362	289
	9	103	81
	10	17	13
Alwatan	6	3024	2605
	7	2152	1525
	8	1072	686
	9	315	134
	10	38	24
Akhbar-Alkhaleej	6	1997	1948
	7	1194	1111
	8	529	488
	9	145	124
	10	22	18

Table 5.2: The degree to which the same features emerged from different feature selection trials

token in contrast with the word (أجهزة) *devices*. Table 5.5 illustrates possible forms of the word *Poet* (شاعر).

It is obvious that such words may appear in the selected features because we did not perform light stemming. As we previously menstioned, Arabic Light stemmer (light10) does not affect the meaning of words. It only strips off the conjunction "Wa" " "و" , some prefixes like:

(ف ، ك ، ل ، ب ، ال ، وال ، بال ، كال ، فال ، لل)

and some suffixes such as:

(ها ، ون ، وا ، ين ، ان ، يه ، ية ، ا ، ة ، ه) [31].

Word	Meaning
الأطفال	The children
الإبداع	The creativity
الانتخابات	The elections
الجل	The gel
الشاعر	The poet
الفيروس	The virus
المهرجان	The festival
النفط	The oil
باحثون	Researchers (masculine)
بارما	Parma (Not Arabic word)
دراسة	Study
دريدا	Derrida (Name)
رياضات	Mathematics
قنديل	Candle
مراكش	Marakech (Name)
وفرنسا	and France
ومعلوم	and known

Table 5.3: Common features in 10 trials Alj-News using BPSO/KNN

5.4 Effect of using normalization and light stemming on BPSO/SVM performance

This section investigates the effect of normalization and light stemming on the classification performance of SVM, NB and C4.5 using BPSO/SVM as a feature selection method. All experiments were repeated after adding normalization and light stemming operations to the text pre-processing steps.

Table 5.6 presents the obtained results after adding normalization and light stemming to the pre-processing. In general, normalization and light stemming have led to slight improvement in some cases while in others, it affects the classification accuracy of the classifiers slightly.

Word	Meaning
أنسجة	Tissues
أوين	Owen (Not Arabic word)
الإسرائيلي	The Israeli
التجارة	The trade
التخطيط	The planning
التونسية	The Tunisian
الثقافة	The culture
السلاح	The weapon
الفرنسية	The French
المشتقات	The derivatives
المهرجان	The festival
سارس	Sars (Not Arabic word)
يسرا	Yusra (Name)

Table 5.4: Common features in 10 trials Alj-News using BPSO/SVM

Figure 5.6 shows the effect of applying normalization and light stemming in the pre-processing stage. For the *Alwatan* dataset, as can be seen from Figure 5.6, normalization and light stemming have improved the performance of C4.5 and SVM slightly while the average accuracies of BN are same. Standard T-test was applied (one-tailed), with significant level $p < 0.1$, and found that the difference between the results for C4.5 is significant ($p = 0.00107$) while in the case of NB, the difference is not significant. For SVM, the difference is significant ($p < 0.08$).

For the *Alj-News* dataset, as in Figure 5.6, generally, normalization and light stemming have resulted in a better performance for C4.5 and NB algorithms. The average accuracies of SVM are similar. The highest improvement is for the NB classifier. For both SVM and C4.5, the differences are not significant while in the case of NB, the difference is statistically significant ($p = 0.00000349$).

Figure 5.6 shows the average performance of SVM, NB and C4.5 on Akhbar-

Word	Meaning
الشاعر	The poet
وشاعر	and poet
كالشاعر	Like the poet
للشاعر	For the poet
بالشاعر	By the poet
والشاعر	and the poet
كشاعر	As a poet

Table 5.5: Different forms of the word (*poet*) in Arabic

Dataset	C4.5	NB	SVM
Alj-News	**0.7575**	**0.838**	**0.9027**
Std. Dev.	0.02621	0.01105	0.01228
P-value	0.14677	0.00000349	0.43296
Alwatan	**0.8366**	**0.9101**	**0.9694**
Std. Dev.	0.01693	0.0175	0.00851
P-value	0.00107	0.4834	0.07798
Akhbar-Alkhaleej	**0.786**	**0.8553**	**0.8857**
Std. Dev.	0.01319	0.01577	0.01075
P-value	0.45084	0.04904	0.12362

Table 5.6: Classification accuracy of SVM, NB and C4.5 on the three datasets (with normalization and light stemming) using BPSO/SVM

Alkhaleej dataset. For C4.5 and NB classifiers, it seems that the performance with applying normalization and light stemming have led to a slight improvement while in the case of SVM classifier, normalization and light stemming have decreased the average accuracy from 0.891 to 0.885 which is small. For C4.5, the difference is not significant while for NB, the difference is significant ($p < 0.05$); however for SVM, the difference is not significant.

Standard T-test (one-tailed) with significant level by$p < 0.1$ (confidence level 90%) was applied to see which method is better (with or without normalization and light stemming using BPSO/SVM for feature selection). Considering the full sets

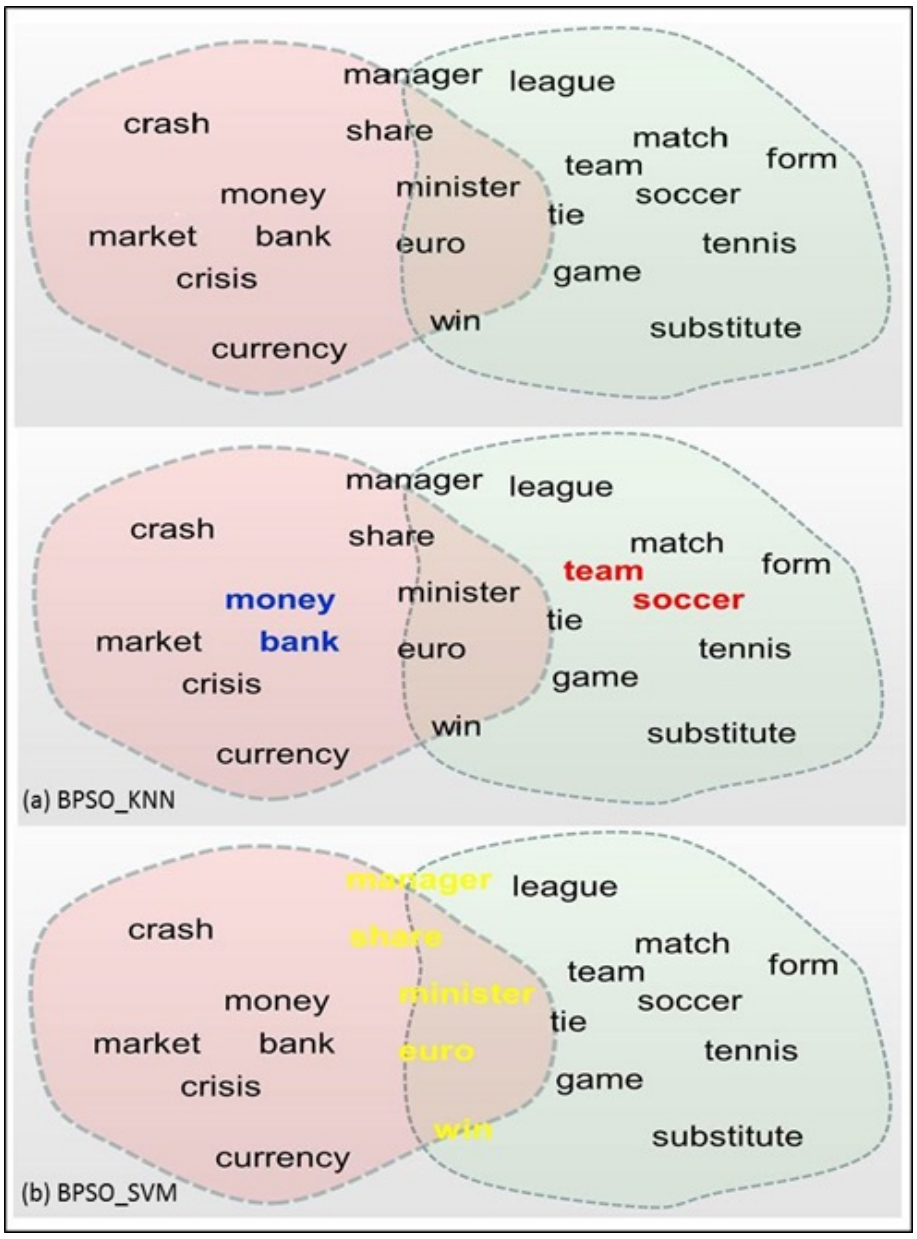

Figure 5.5: Selected feature by BPSO/KNN and BPSO/SVM

of results using BPSO/SVM, on Alwatan dataset, a one tailed-T-test shows that BPOS/SVM with normalization and light stemming is significantly better than BPSO/SVM without normalization and light stemming ($p < 0.003$) with confidence level 99%. Also, on Alj-News datasets, BPSO/SVM with normalization and light stemming is significantly better than BPSO/SVM without normalization and light stemming ($p < 0.007$) with confidence level 99%. In case of Akhbar-Alkhaleej dataset, BPSO/SVM without normalization and light stemming has a better average than BPSO/SVM with normalization and light stemming, but a one-tailed T-test reveals that this difference is not significant, and we cannot be sure which of BPSO/SVM with normalization and light stemming or without normalization and

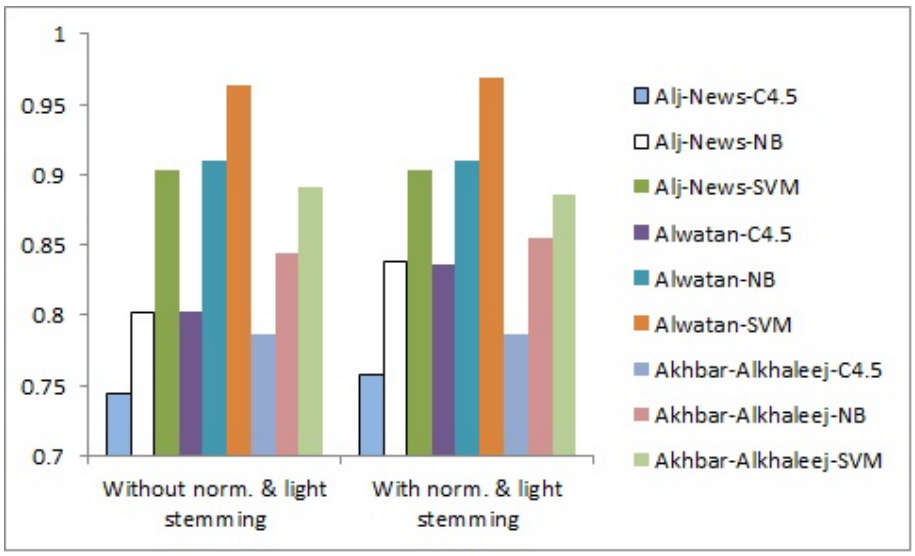

Figure 5.6: Classification accuracy of SVM, NB and C4.5 using BPSO/SVM

light stemming is the best method.

5.5 Comparison between BPSO with KNN and BPSO with SVM after applying normalization and light stemming

Figure 5.7 shows the difference in the performance between BPSO with KNN and BPSO with SVM after normalizing some Arabic letters and performing Arabic light stemming during text pre-processing. Standard T-tests were applied (one-tailed), with significance level ($p < 0.1$). On Alwatan dataset, as shown in Figure 5.7, the results of C4.5 for BPSO/SVM are superior to those of BPSO/KNN ($p < 0.07$); however the corresponding results for SVM and NB are not significant. On *Alj-News* dataset, as in Figure 5.7, the results of SVM for BPSO/KNN are superior to those of BPSO/SVM ($p < 0.05$) while the corresponding results for C4.5 and NB show no significant difference. On *Akhbar-Alkhaleej* dataset, as presented in Figure 5.7, the results of the three classifiers show no significant difference.

Standard T-test (one-tailed) with significant level by $p < 0.1$ (confidence level 90%) was applied to see which method is better after applying normalization and light stemming. On the three datasets, one-tailed T-tests reveal that the differences

are not significant, and we cannot be sure which of BPSO/SVM or BPSO/KNN is the best method.

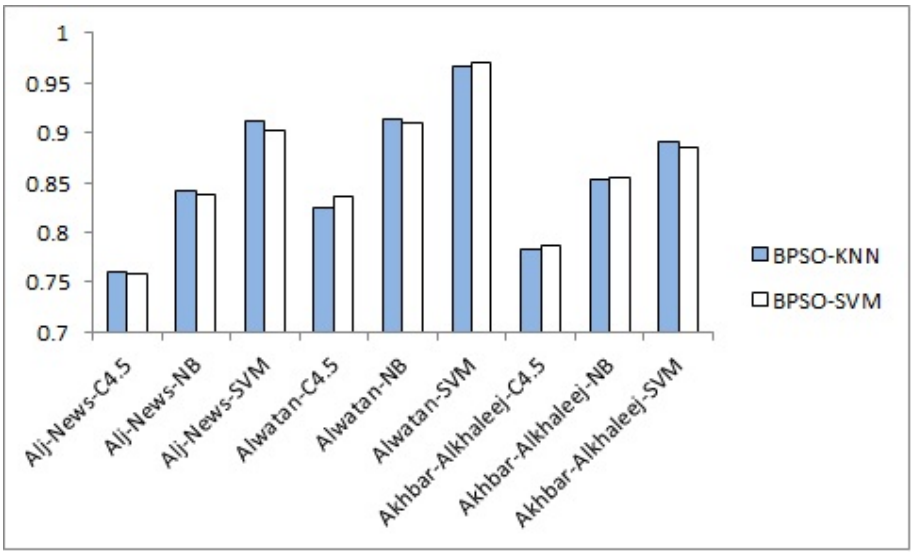

Figure 5.7: Performance of BPSO/SVM and BPSO/KNN after applying normalization and light stemming

Similar to the comparison between BPSO/SVM and BPSO/KNN when we did not use normalization and light stemming, Table 5.7 presents the degree to which individual features repeatedly emerged in different trial runs on the same data with the use of normalization and light stemming in the pre-processing. On *Alwatan* dataset, Table 5.7 tells us there were 337 features (words) that each occurred in precisely 8 of the 10 trials of BPSO/SVM, while the corresponding number for BPSO/KNN was 623. We also see that, for BPSO/SVM, there were 13 features present in the result of every trial, while the corresponding figure for BPSO/KNN was 22. Overall, feature sets from different trial runs of BPSO/KNN tend to have a greater overlap with each other than those emerging from BPSO/SVM trials.

5.6 Summary

This chapter described BPSO/SVM as a feature selection method for Arabic text categorization (TC). It was compared with BPSO/KNN, and statistical evidence has indicated that BPSO/SVM should be favoured. We also speculated on factors that might influence the relative performances of BPSO/SVM and BPSO/KNN. Analysis shows that each approach relies on a different core set of features. We put forward a

Dataset	No. of Trials	BPSO/KNN	BPSO/SVM
Alj-News	6	848	768
	7	600	512
	8	255	219
	9	88	66
	10	20	21
Alwatan	6	1658	1422
	7	1219	855
	8	623	337
	9	170	91
	10	22	13
Akhbar-Alkhaleej	6	1048	1167
	7	779	657
	8	408	292
	9	130	73
	10	28	12

Table 5.7: The degree to which the same features emerged from different feature selection trials (Normalization and light stemming)

simple argument to explain this in terms of the different classification models used by SVM and KNN, and this further suggests BPSO/SVM may be favoured when the categories tend to be quite close. This could point towards useful guidelines for choosing FS methods in this area. For example, if categories tend to be quite distinct, then BPSO/KNN (or similar) may be favourable simply because of the ability to more speedily determine a feature subset. However, if accuracy in distinguishing between close categories is important, we might accept the computational cost of BPSO/SVM.

In addition, the effect of applying normalization and light stemming in the preprocessing on the performance of SVM, NB and C4.5 with features selected by BPSO/SVM was also examined. Results show that applying normalization and light stemming in Arabic TC leads to a significant reduction in the number of required features for text representation. In terms of classification accuracy, statistical

evidence revealed that including normalization and light stemming yields similar results or even better in some cases i.e. Naive Bayes with BPSO/SVM on Alj-News and Akhbar-Alkhaleej datasets with a fewer number of features than no normalization and stemming case.

Chapter 6

Statistical phrase based text representation for Arabic Text Classification

This chapter investigates the impact of using statistical phrases of length two instead of single words for representing text documents on the performance of both feature selection methods for the Arabic TC task. In terms of classification accuracy, this chapter also presents the results of BPSO/KNN and BPSO/SVM with three ML classifiers using a combination of single words and phrases for text representation.

Most text representation used in text mining, information retrieval and related applications is the bag of words approach (BOW). In an attempt to improve the classification accuracy of TC classifiers, a number of researchers have tried to use phrases instead of or beside single words as features to represent text documents [10, 11].

In general, two different types of phrases have been proposed and investigated:

- *Syntactic phrase:*
  Syntactical phrases such as noun phrases and verb phrases can be extracted from text documents based on syntactical rules [10, 12].
- *Statistical phrase:*
  Following stop-words removal, statistical phrase is defined as a sequence of n words occurring consequently in the context [10, 11].

The advantage of using phrases for text representation is that phrases have larger meaning than single words [11, 12]. Unfortunately, in terms of classification accuracy, most of the work done using different forms of phrases for text representation did not show better or encouraging results in comparison with single word representation [10, 11].

6.1 Extracting phrases from Arabic texts

In this work, the effect of using statistical phrases as document indexing method for the Arabic text categorization problem was investigated. To extract statistical phrases of length two (bigrams) from the training set, all Arabic text documents have been pre-processed according to the following steps:

- Conversion to UTF-8 encoding.
- Remove hyphens, punctuation marks, numbers, digits, non-Arabic letters and diacritics.
- Remove stop words.
- Perform Normalization of Arabic letters using [31, 32].
- Perform Arabic word light stemming using (light10) [31, 32].
- Extract stemmed and ordered bigrams (ordered as they appear in the text).
- Eliminate rare phrases (phrases that occur less than five times in the dataset).
- The standard Vector Space Model (VSM) was applied to represent Arabic texts [4] and TFIDF was used for the term weighting.

The developed pre-processing software generates a training set based on all distinct features (bigrams) to be used by the feature selection approach (BPSO) to select the best subset of features. Then, the training and test sets were built using only the optimal subset of features. Note that in our experiments all the pre-processing and weighting was performed separately for the training set and the test set.

The pseudo code for extracting the distinct features (phrases of length two) from a given training corpus is as follows:

```
Define List_phrases_each_file<phrase,count>
Define List_of_Distinct_phrases<phrase,count>

for each text_file do
{
Temp_List=Tokinze()
Temp_List=Remove_Stop_Words(Temp_List)
Temp_List=Normalize_Words(Temp_List)
Temp_List=LightStem(Temp_List)

// To form phases (Bigrams)
for each token Temp_List do
  {
  Temp_list[i] = temp_list[i]+"-"+ temp_list [i+1]
  }
for each phrase Temp_List do
  {
if(!File_Check(phrase,List_phrases_each_file[])
List_phrases_each_file[].add(phrase)
else
List_phrases_each_file[].Increment(phrase)
  }
}
for each List_phrases_each_file do
{
for each phrase in List_phrases_each_file do
 {
if(!Check_phrases(List_of_Distinct_phrases,phrase)) then
  {
Add_phrases_Count(List_of_Distinct_phrases,phrase,count)
```

```
else
Add_count_only(List_of_Distinct_phrases,phrase,count)
  }
 }
}
```

The output of the above procedure is a list of distinct phrases (features) with their counts. The next step is to remove rare phrases (phrases that appear less than five times in the corpus).

After rare features (phrases) elimination, the last step is to calculate the TFIDF of each distinct feature for each document in the corpus. In this way, each document is represented as a vector of features (phrases) e.g. $(f_{i1}, f_{i2}, ..., f_{iN})$ where i refers to the i_{th} document in the dataset and N is the number of phrases that represent documents.

6.2 Experiments

Two Arabic datasets were used to evaluate the effect of phrase based Arabic text representation on the Arabic text categorization problem. After pre-processing Arabic datasets, the numbers of distinct features (phrases) found in the separate datasets were as shown in Table 6.1. These directly represent the sizes of the BPSO particles. Finally, each document in each dataset was trnasformed into a feature (phrase) vector.

Dataset	Distinct features (bigrams) from the training set
Alj-News	3446
Akhbar-Alkhaleej	7586

Table 6.1: Distinct features (phrases) from the training set

In Table 6.2, we show the overall summarized results on the test set in terms of F-measure (recall this is an average of 10 complete trials of the training/test process) using phrases for text representation with the BPSO/KNN feature selection

method. On both datasets, in comparison with BOW representation, it is clear from Figure 6.1 that using phrases as features for the Arabic TC task decreases the classification accuracy of TC algorithms. On *Alj-News* dataset, as shown in Table 6.2, NB outperformed SVM while on *Akhbar-Alkhaleej*, standard T-test (one-tailed) with significant signalling by $p < 0.1$ were applied and showed no significant difference between the average accuracies of SVM and NB.

Dataset	J48	NB	SVM
Alj-News	**0.5118**	**0.7217**	**0.6753**
Std. Dev	0.03318	0.01577	0.02789
Akhbar-Alkhaleej	**0.7419**	**0.7653**	**0.7604**
Std. Dev	0.03036	0.01761	0.01487

Table 6.2: Classification accuracy of C4.5, NB and SVM using Phrases only (BPSO/KNN)

In Table 6.3, we show the overall summarized results on the test set in terms of F-measure (recall this is an average of 10 complete trials of the training/test process) using phrases for text representation with the BPSO/SVM feature selection method. As shown in Figure 6.2, it is clear that the performances of C4.5, NB and SVM using BOW as features for Arabic TC task are superior to those using phrases. On *Alj-News* dataset, it can be seen that NB outperforms SVM while on *Akhbar-Alkhaleej*, standard T-tests (one-tailed) with significance level of $p < 0.1$ were applied and revealed that there is no significant difference between the average accuracies of SVM and NB.

For more detailed views of the results using phrases of length two for text representation, we will show details of one of the ten trials on each of the two datasets using BPSO/SVM as the feature selection technique. Clearly, NB was able to classify most accurately. Tables from 6.4 to 6.9 set out more detailed views of the results on each of the two datasets, showing mean values for precision, recall and F-measure for each category in the dataset in question, where the averages are weighted according to the numbers of documents in each category. Tables 6.4, 6.5, and 6.6 respectively show the results for SVM, Naive Bayes, and C4.5 on the *Alj-News* dataset, while

Dataset	J48	NB	SVM
Alj-News	**0.5125**	**0.7136**	**0.6864**
Std. Dev.	0.03074	0.01944	0.0238
P-value	0.48075	0.16018	0.17571
Akhbar-Alkhaleej	**0.7439**	**0.7798**	**0.7751**
Std. Dev.	0.02867	0.017	0.02343
P-value	0.44065	0.03868	0.05713

Table 6.3: Classification accuracy of C4.5, NB and SVM using Phrases only (BPSO/SVM)

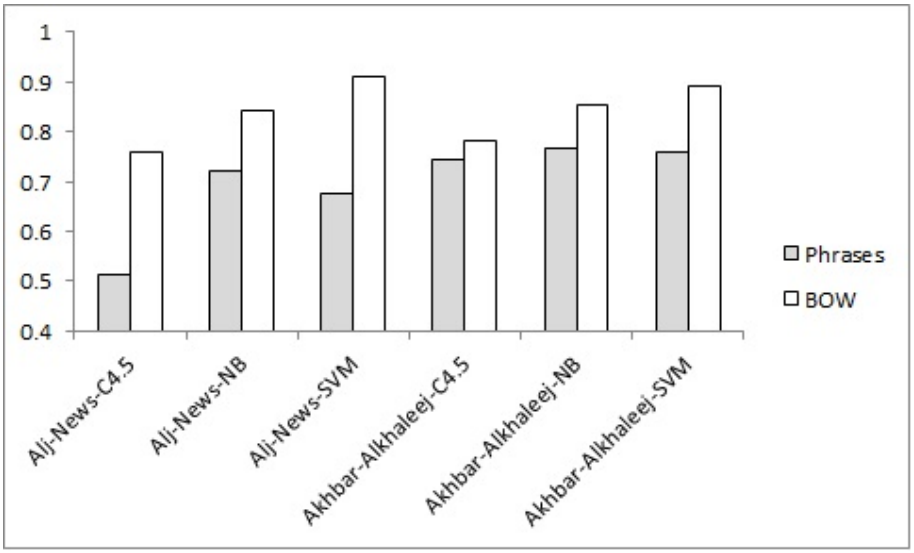

Figure 6.1: Classification accuracies of C4.5, NB and SVM using BPSO/KNN

the sequence is repeated for the *Akhbar-Alkhaleej* in Tables 6.7 to 6.9.

For *Alj-News* dataset, as shown in Tables 6.4, 6.5 and 6.6 respectively, it is clear that NB outperformed both SVM and C4.5. As in Table 6.4, Sport and Science categories have the highest precision and recall in contrast with other categories. This means that SVM correctly predicted the categories of most instances related to those categories. In addition, the category with the lowest precision is Art. This could be interpreted that there is overlap between the Art category and other categories, most likely Economics and Politics. For NB classifier, as in Table 6.5, the precision of Sport category is 1. This indicates that the false positive rate for this category is 0. In other words, NB did not assign any document from all other categories to Sport. Art and Politics have the lowest precisions. This means that some docu-

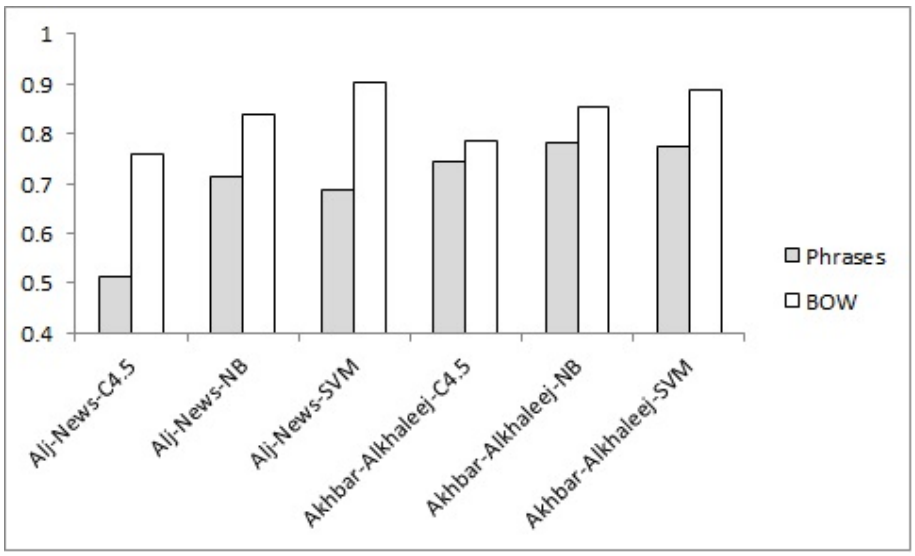

Figure 6.2: Classification accuracies of C4.5, NB and SVM using BPSO/SVM

ments from other categories are incorrectly labelled as Art or Politics. In addition, the Politics category has the highest recall. This reflects the fact that documents from other categories were classified by NB as Politics.

Class	**Precision**	**Recall**	**F-Measure**
Sport	0.804	0.75	0.776
Art	0.488	0.667	0.563
Science	0.932	0.683	0.788
Politics	0.591	0.65	0.619
Economics	0.75	0.65	0.696
Weighted Avg.	0.713	0.68	**0.689**

Table 6.4: Detailed Accuracy by Class for SVM Classifier on Alj-News Dataset

For C4.5, as shown in Table 6.6, the precision of the Sport category is 1; this means that the false positive FP rate is 0. It can also be noticed that it is a challenge for C4.5 classifier in distinguishing between documents in the political category and documents of other categories. In general, on *Alj-News* dataset, NB is the best one using phrases for text representation while C4.5 is the worst.

Tables 6.7, 6.8 and 6.9 show the performance of SVM, NB and C4.5 respectively on *Akhbar-Alkhaleej* dataset. It is clear that NB outperformed SVM and C4.5. It can be seen that the three used classifiers have high precisions on the Sport cat-

Class	Precision	Recall	F-Measure
Sport	1	0.817	0.899
Art	0.577	0.75	0.652
Science	0.915	0.717	0.804
Politics	0.543	0.833	0.658
Economics	0.912	0.517	0.66
Weighted Avg.	0.789	0.727	**0.734**

Table 6.5: Detailed Accuracy by Class for NB Classifier on Alj-News Dataset

Class	Precision	Recall	F-Measure
Sport	1	0.583	0.737
Art	0.36	0.667	0.468
Science	0.96	0.4	0.565
Politics	0.471	0.8	0.593
Economics	0.815	0.367	0.506
Weighted Avg.	0.721	0.563	**0.574**

Table 6.6: Detailed Accuracy by Class for C4.5 Classifier on Alj-News Dataset

egory which means that all of them managed to discriminate well between other categories and Sport and also other categories from Int. News category. The values of precisions for Economy and local news indicate that it is a challenge for the three classifiers to separate either Local news or Economy from Sport and Int. News categories.

Class	Precision	Recall	F-Measure
Economy	0.711	0.491	0.581
Int. News	0.907	0.672	0.772
Local News	0.679	0.896	0.772
Sport	0.958	0.802	0.873
Weighted Avg.	0.793	0.77	**0.767**

Table 6.7: Detailed Accuracy by Class for SVM Classifier on Akhbar-Alkhaleej Dataset

Tables 6.10 to 6.15 show the confusion metrics. The confusion between topics is

Class	Precision	Recall	F-Measure
Economy	0.755	0.673	0.712
Int. News	0.907	0.845	0.875
Local News	0.724	0.875	0.792
Sport	0.909	0.698	0.789
Weighted Avg.	0.806	0.793	**0.793**

Table 6.8: Detailed Accuracy by Class for NB Classifier on Akhbar-Alkhaleej Dataset

Class	Precision	Recall	F-Measure
Economy	0.477	0.564	0.517
Int. News	0.946	0.603	0.737
Local News	0.636	0.764	0.694
Sport	0.897	0.709	0.792
Weighted Avg.	0.728	0.691	**0.697**

Table 6.9: Detailed Accuracy by Class for C4.5 Classifier on Akhbar-Alkhaleej Dataset

generally understandable given the levels of similarity between the topics in different cases. From Table 6.10, it can be seen that SVM incorrectly assigned documents from other categories to Art e.g. 13 and 10 documents from Politics and Sport respectively. For the Science category, only 3 documents, 2 related to Art and 1 related to Sport, were wrongly predicted as Science. In Table 6.11, we can see that NB did not classify any documents from other categories as Sport. The raw indicates that NB has wrongly predicted about half of the Economics documents as Politics and Art. As in Table 6.12, C4.5 has the same difficulty in distinguishing Politics and Art from other categories.

As shown in Table 6.13, on *Akhbar-Alkhaleej* dataset, 264 out of 343 test instances were correctly classified by SVM. It is clear from column 3 that it is hard for SVM to discriminate between local news and other categories documents. Table 6.14 tells us that NB found difficulty in separating categories like Sport and Economy from Local news category e.g. 26 Sport documents were predicted as Local news however 272 out of 343 were correctly classified by NB. If we look at Table

a	b	c	d	e	⟸ classified as
45	10	1	4	0	a = sport
4	**40**	2	8	6	b = art
1	9	**41**	7	2	c = science
3	13	0	**39**	5	d = politics
3	10	0	8	**39**	e = economic

Table 6.10: Confusion Matrix (SVM on Alj-News Dataset)

a	b	c	d	e	⟸ classified as
49	4	1	6	0	a = sport
0	**45**	2	12	1	b = art
0	7	**43**	10	0	c = science
0	8	0	**50**	2	d = politics
0	14	1	14	**31**	e = economic

Table 6.11: Confusion Matrix (NB on Alj-News Dataset)

6.15, it is clear that Local news category is behind most of the misclassifications done by C4.5. It can be said that it is a challenge for a machine learning classifier to correctly distinguish between categories that are close in nature to each other e.g. Local news and Sport.

6.3 Comparison between BPSO with KNN and BPSO with SVM using phrases for text representation

Tables 6.2 and 6.3 showed the results of BPSO/KNN and BPSO/SVM using phrases for text representation. Standard T-test (one-tailed) with significance level of $p < 0.1$ was applied and revealed the following findings. On *Alj-News* dataset, none of the corresponding pairwise results are significant, while on *Akhbar-Alkhaleej* dataset, the NB results with BPSO/SVM are superior to those of BPSO/KNN ($p < 0.04$); also, the results of SVM with BPSO/SVM outperformed those of BPSO/KNN ($p < 0.06$); however the corresponding results of C4.5 are not significant.

a	b	c	d	e	⟸ classified as
35	15	0	9	1	a = sport
0	**40**	1	17	2	b = art
0	24	**24**	10	2	c = science
0	12	0	**48**	0	d = politics
0	20	0	18	**22**	e = economic

Table 6.12: Confusion Matrix (C4.5 on Alj-News Dataset)

a	b	c	d	⟸ classified as
27	2	26	0	a = Economy
0	**39**	19	0	b = International
10	2	**129**	3	c = Local
1	0	16	**69**	d = Sport

Table 6.13: Confusion Matrix (SVM on Akhbar-Alkhaleej Dataset)

Standard T-test (one-tailed) with significant level by $p < 0.1$ (confidence level 90%) was applied to see which method is better using phrases only. Considering the sets of results using phrases only with BPSO/KNN or BPSO/SVM, on the Alj-News dataset, a one-tailed T-test reveals that the difference is not significant, and we cannot be sure which of BPSO/SVM or BPSO/KNN is the best method. In case of Akhbar-Alkhaleej dataset, ($p < 0.08$), this means 92% confidence that BPSO/SVM is really better than BPSO/KNN.

6.4 Using a combination of phrases and BOW for text representation

This section investigates the effect of using a combination of bag-of-words and phrases (bigrams) (BOW+ phrases) on the classification accuracy of the classifiers. Table 6.16 presents the results using phrases and single words together for term indexing. To compare those results with the case of using single terms (BOW), standard T-test was applied (one-tailed) with significant signalling by $p < 0.1$, and found the following. For the BPSO/KNN approach, on *Alj-News* dataset, Figure

a	b	c	d	⟸ classified as
37	2	14	2	a = Economy
1	**49**	8	0	b = International
11	3	**126**	4	c = Local
0	0	26	**60**	d = Sport

Table 6.14: Confusion Matrix (NB on Akhbar-Alkhaleej Dataset)

a	b	c	d	⟸ classified as
31	1	22	1	a = Economy
4	**35**	19	0	b = International
27	1	**110**	6	c = Local
3	0	22	**61**	d = Sport

Table 6.15: Confusion Matrix (C4.5 on Akhbar-Alkhaleej Dataset)

6.3 reveals that the results of BOW representation for NB and C4.5 are slightly better than using bag-of-words and phrases. The T-test shows that the differences for the three classifiers are not statistically significant. As can be seen from Figure 6.3, on *Akhbar-Alkhaleej* dataset, although the results of BPSO/KNN with BOW and phrases are superior to those of BPSO/KNN with BOW only, the differences for the three classifiers are not statistically significant.

Dataset	C4.5	NB	SVM
Alj-News	**0.7559**	**0.8337**	**0.9116**
Std. Dev.	0.00871	0.01299	0.00849
P-value	0.22112	0.16963	0.45753
Akhbar-Alkhaleej	**0.7915**	**0.8604**	**0.8992**
Std. Dev.	0.01936	0.00867	0.01208
P-value	0.14954	0.15146	0.12362

Table 6.16: Classification accuracy of C4.5, NB and SVM with BPSO/KNN using combination of bag-of-words and phrases

For the BPSO/SVM approach, Table 6.17 shows the results. Standard T-test was applied (one-tailed), with significant signalling by $p < 0.1$, and found the following. On *Alj-News*, as shown in Figure 6.4, the SVM results for BPSO/SVM with

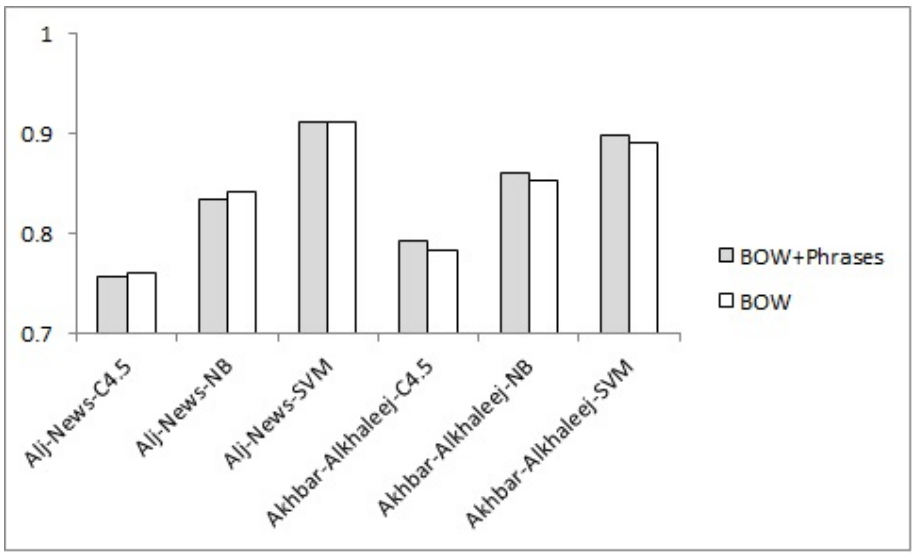

Figure 6.3: Classification accuracy of C4.5, NB and SVM with BPSO/KNN using combination of bag-of-words and phrases

BOW and phrases are superior to those of BPSO/SVM with BOW only ($p < 0.02$); however, the corresponding results for NB and C4.5 are not significant. As can be seen in Fgure 6.4, on *Akhbar-Alkhaleej*, SVM results for BPSO/SVM with BOW and phrases are superior to those of BPSO/SVM with BOW only ($p < 0.09$), while the corresponding results for NB and C4.5 are not significant.

Standard T-test (one-tailed) with significant level by $p < 0.1$ (confidence level 90%) was applied to see which method is better (BOW only or a combination of BOW and phrases). Considering the full sets of results using combination of BOW and phrases as features, on Alj-News datasets, one-tailed T-tests reveal that the differences are not significant, and we cannot be sure which of BOW only or BOW plus phrases is the best. In case of Akhbar-Alkhaleej dataset, using BPSO/KNN, one tailed T-test shows ($p < 0.03$) that using BOW plus phrases is better than using BOW only (confidence level 98%). In case of use BPSO/SVM, one tailed T-test shows ($p < 0.05$) that using BOW plus phrases is better than using BOW only (confidence level 95%).

Dataset	C4.5	NB	SVM
Alj-News	**0.7546**	**0.8347**	**0.9151**
Std. Dev.	0.02196	0.01719	0.01229
P-value	0.39585	0.30842	0.01834
Akhbar-Alkhaleej	**0.7901**	**0.8567**	**0.9007**
Std. Dev.	0.01661	0.01752	0.02985
P-value	0.27452	0.42657	0.08116

Table 6.17: Classification accuracy of C4.5, NB and SVM with BPSO/SVM using combination of bag-of-words and phrases

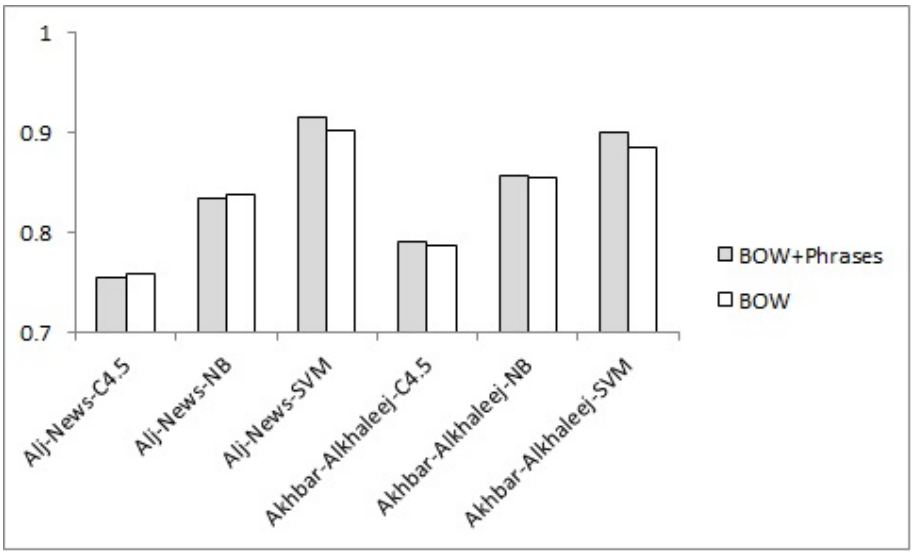

Figure 6.4: Classification accuracy of C4.5, NB and SVM with BPSO/SVM using combination of bag-of-words and phrases

6.5 Comparison between BPSO with KNN and BPSO with SVM using BOW and phrases for text representation

It can be noticed from Tables 6.18 and 6.19 that feature sets from different runs of BPSO/KNN have a greater overlap than those from BPSO/SVM using BOW only or combination of BOW and phrases for text representation while when phrases alone are used, features from BPSO/SVM have a greater overlap than those from BPSO/KNN. We also report that the intersections between frequently occurring BPSO/KNN and BPSO/KNN features were always zero or very small using BOW,

phrases, or a combination of both.

Dataset	No. of Trials	BPSO/KNN	BPSO/SVM
Alj-News	8	227	287
	9	57	98
	10	5	17
Akhbar-Alkhaleej	8	442	540
	9	95	158
	10	20	46

Table 6.18: The degree to which the same features emerged from different feature selection trials (Phrases Only)

Dataset	No. of Trials	BPSO/KNN	BPSO/SVM
Alj-News	8	560	411
	9	169	121
	10	26	23
Akhbar-Alkhaleej	8	848	782
	9	231	207
	10	36	22

Table 6.19: The degree to which the same features emerged from different feature selection trials (BOW+ Phrases)

Tables 6.20 and 6.21 present the number of selected single words and phrases from different runs of BPSO/KNN and BPSO/SVM on *Alj-News* and *Akhbar-Alkhaleej* Arabic datasets. On *Akhbar-Alkhaleej* dataset, it seems that the number of selected phrases is more than the number of single words (8 and 9 trials) while on *Alj-News* dataset, the number of single words is more than phrases.

6.6 Summary

This chapter investigated the impact of using statistical phrases of length two on the classification accuracy of Arabic text classification using two feature selection

No. of Trials	BPSO/KNN		BPSO/SVM	
	Words	Phrases	Words	Phrases
8	288	272	229	182
9	102	67	79	42
10	17	9	17	6

Table 6.20: Number of selected single words and phrases from different runs of BPSO/KNN and BPSO/SVM on Alj-News dataset

No. of Trials	BPSO/KNN		BPSO/SVM	
	Words	Phrases	Words	Phrases
8	349	499	339	443
9	86	145	79	128
10	21	15	13	9

Table 6.21: Number of selected single words and phrases from different runs of BPSO/KNN and BPSO/SVM on Akhbar-Alkhaleej dataset

methods BPSO/KNN and BPSO/SVM. Experiments were conducted on two Arabic datasets with three machine learning classifiers. In comparison with results using BOW, using statistical phrases for text representation decreases the accuracy of Arabic text classifiers significantly. It was observed that NB classifier works better than SVM when phrases were used for text representation.This chapter also examined the benefit of using a combination of single words and phrases for Arabic text representation. In comparison with the bag of words approach for text representation, results of BPSO/KNN and BPSO/SVM with SVM, NB and C4.5 on two datasets reveal that applying single words and phrases for term indexing leads to a slight statistically significant improvement in the performance of SVM while for NB and C4.5, the differences are not statistically significant.

Chapter 7

Conclusion

7.1 Summary

This research proposed feature selection techniques for Arabic document classification. Experimental results showed that the developed techniques are promising in this domain. These techniques could be useful for many application related to the Arabic natural language processing field. For example, filtering e-mails written in Arabic based on the users interest (e. g. spam filtering), support tools that post-process the results of Arabic search engines and organizing large collections of Arabic web pages under hierarchal categories. When Arabic web pages are organized in this way, it is faster for an Arabic web search engine to start navigating the hierarchy of categories and then limiting its search in the category that contains the required information.

The first developed feature selection approach is based on the well-known optimization algorithm Binary Particle Swarm Optimization technique in conjunction with K Nearest Neighbour classifier. Three Arabic document collections were used to test this approach, and three well-known classification algorithms SVM, Naive Bayes and C4.5 decision tree learning were used to classify Arabic documents using features selected by this approach. The obtained results suggest that the proposed method is effective. It led to values for classification accuracy and F1-measure that compare well with those reported in the related work; i.e. the tenfold-cross validation performance of SVM, when applied to holdout dataset restricted to feature subsets emerging from the feature selection process, ranged from 89% and 96% while

for Naive Bayes it is in the range 80% and 89%. The lowest accuracy values were associated with *Akhbar-Akhaleej* dataset. It is not surprising because the nature of categories like local news and international news can often be both about sport and the economy. It is quite a challenge for any automated method to correctly predict the labelled categories. In addition, the effect of integrating the normalization of some Arabic letters and performing light stemming in the pre-processing were also studied. Normalization and light stemming have led to a significant reduction in the number of distinct features e.g. in the case of *Alwatn* dataset, the number of distinct features was reduced from 12282 to 7062 while for *Alj-News* dataset it dropped from 5329 to 3763. In terms of classification accuracy, for most of the evaluation cases, normalization and light stemming improved the classification accuracy of the classifiers slightly. For instance, the average classification accuracy of Naive Bayes on Ali-News dataset increased from 80% to 84% and on *Akhbar-Alkhaleej* dataset from 83% to 85%. It is clear that the results achieved by SVM, as well as NB, overall demonstrate that BPSO/KNN performs well as a feature selection technique for the Arabic document classification task.

Our second idea is to replace KNN classifier with SVM. The classification results obtained by BPSO/SVM was compared with BPSO/KNN results, and statistical evidence indicates that BPSO/SVM should be favored. We found that each approach relies on a different core set of features. This could point towards useful guidelines for choosing FS approaches in this area. For example, if categories tend to be quite distinct, then BPSO/KNN (or similar) may be favorable simply because of the ability to more speedily determine a feature subset. However, if accuracy in distinguishing between close categories is important, BPSO/SVM is better becuase of its ability to discriminate between close categories. In addition, the effect of applying normalization and light stemming in the pre-processing on the performance of SVM, NB and C4.5 with features selected by BPSO/SVM was also examined. Results show that applying normalization and light stemming in Arabic TC leads to a significant reduction in the number of required features for text representation. In terms of classification accuracy, statistical evidence revealed that including normalization and light stemming yields similar results or even better in some cases i.e. Naive Bayes with BPSO/SVM on *Alj-News* and *Akhbar-Alkhaleej* datasets with fewer features performs better than no normalization and light stemming case.

The usefulness of using phrases instead of single words for representing text documents in Arabic TC was also investigated. Statistical phrases of length two were extracted and used for Arabic text representation. In terms of classification accuracy, although phrases have a larger meaning than single words, for the Arabic TC problem, results prove that using phrases for text representation decreases the classification accuracy of SVM, NB and C4.5 significantly when those classifiers were applied with either BPSO/KNN or BPSO/SVM as feature selection methods for Arabic TC.

In case of using combinations of single words and phrases for text representation, in comparison with bag of words approach, results of BPSO/KNN and BPSO/SVM with SVM, NB and C4.5 on two datasets showed that using single words and phrases for term indexing improves the performance of SVM classifier slightly.

7.2 Accomplishments

The main accomplishments of this thesis can be listed as follows:

- This work demonstrates successful document classification in the context of Arabic documents (although previous work has demonstrated text classification in Arabic, the datasets used and the experimental setup have not been revealed).
- We demonstrate a combination of Binary PSO and K nearest neighbour that performs well in selecting good sets of features for this task. Comparison with related work in this area is not currently possible, since either the datasets used in an associated publication are not available, or, where they are available, we have been unable to discover the way the data was split into training and/or validation and/or test data in the comparative results. Therefore another achievement of this work is to make our datasets available, with the latter issues clarified, to support continuing work on this topic.
- This work also proposes a combination of Binary PSO and Support Vector Machine that performs well in selecting good sets of features for this task.

- The proposed FS method BPSO with SVM was compared with KNN based approach. Following a run of either BPSO/KNN or BPSO/SVM, the resulting feature subset is evaluated on a holdout (unseen) dataset. The later evaluation is done using either SVM, Naive Bayes, or C4.5 (using Weka machine learning software). Results suggest that BPSO/SVM may be preferable when the distance between categories is small, since several documents will be close to boundaries while features most often chosen by KNN could be core to a category in the sense they are close to a centre in document vector space.

- The impact of using Arabic light stemming and normalization of some Arabic letters has also been studied. In comparison with not performing normalization and light stemming in the pre-processing of Arabic text documents, it was found that applying light stemming and normalization significantly reduced the number of features needed for Arabic text representation without decreasing the classification accuracy of Arabic TC classifiers.

- Based on the fact that phrases have larger meaning than single words, for Arabic text representation, the benefit of using statistical phrases of length two instead of single words was also examined. In comparison with single words text representation, it was found that using statistical phrases alone as terms for Arabic text classification lowers the classification accuracy of the applied classifiers significantly.

- We also tested the usefulness of using a combination of single words and statistical phrases of length two for text representation. The results show that this combination has improved the performance of SVM for classifying Arabic texts slightly.

7.3 Future work

Possible directions of the future work may include the following ideas:

- The analysis of selected features show that each of BPSO with KNN and BPSO with SVM tend to choose distinct core sets from each other. The preferable features for SVM are those that help define clear boundaries between categories in the transformed feature space. Meanwhile, KNN prefers features which are

central to categories. E.g. consider a document in local news that is close to the boundary with international news. KNN will misclassify this document if its closest neighbours tend to be over the boundary, however SVM will classify it correctly. It can be suggested that SVM features may be chosen according to how well boundaries can be defined, and KNN features may be more biased towards a large distance between documents in different categories. Based on these findings, hybrid approaches may be worth investigation, where (for example) BPSO/KNN may be used to find a feature subset that works well on a distinct partition of categories that aggregates similar categories together, and BPSO/SVM may then be used to find feature subsets specifically to distinguish between the close categories.

- In this work, we use TFIDF for term weighting without normalizing the weights of features. Usually, there are short and long text documents in the corpus. To make all documents have the same level of importance, TFIDF weights can be normalized to equalize the Euclidian length of all feature vectors of documents [24, 101]. We can investigate the impact of applying different term weighting schemes as in [101] such as TF transformation ($log(1 + t_{ij}), t_{ij}$ refers to term i in document j) or TF.IDF transformation (those schemes are provided by Weka software) which may lead to a positive effect on the classification accuracy of the proposed feature selection methods and TC classifiers (SVM, Naive Bayes and C4.5).

- Investigate the usefulness of using different sorts of phrases for Arabic text representation with the proposed feature selections in this work such as noun phrases or a combination of noun phrases and single words. Arabic grammatical rules can be used to extract noun phrases such as 'computer science' and 'Artificial intelligence' to be used as terms for text representation.

- Reference [110] shows better results using concepts rather than BOW for Arabic text representation in Arabic TC problem (concepts were obtained using publically available Arabic WordNet software). In this approach, based on a list of words and concepts (each word has a set of ordered concepts), words with similar meaning are replaced with the most appropriate concept. It would be useful to examine the benefit of using concepts with our proposed feature

selection techniques for Arabic TC.

- Instead of using the traditional form of PSO algorithm (search is based on best global), one can try using different search strategy for PSO such as fully informed PSO (in this method, the particle also interacts with its neighbours not the best global only). Experimental results show that FIPSO can find better solution than traditional PSO with less number of generations [58].

Appendix A

Java Code of Binary Particle Swarm Optimization Algorithm

```
import java.util.Iterator;
import java.util.Random;
import java.util.TreeMap;
public class Particle
{
private double [] current;
private double [] previous;
private double [] velocity;
private double fitness_value;
private int Particle_size;
public static final int Max_Size=80000;
// Constructor
public Particle(int ps)
{
current = new double[Max_Size];
previous = new double[Max_Size];
velocity = new double [Max_Size];
fitness_value=0.0;  //initial fitness value
Particle_size=ps;
}
// This method initializes particles
```

```
// with random positions and velocities
 public void initilize()
{
Random generator = new Random();
double value=0.0;
for(int j=0; j< Particle_size;j++)
{
value=generator.nextDouble()*1;
if(value<0.5)
current[j]=0.0;
else
current[j]=1.0;
 previous[j]=current[j];
value=generator.nextDouble()*1;
 if(value<0.5)
velocity[j]=0.0;
 else
velocity[j]=1.0;
}
}
public int get_particle_size()
 { return Particle_size; }
public double get_Xi(int j)
{ return previous[j]; }
public void print()
{
System.out.println("\nParticle size = "+Particle_size);
System.out.println("Current :");
for(int j=0; j< Particle_size;j++)
{ System.out.print(current[j]+"'"); }
System.out.println("\nprevious :");
for(int j=0;j<Particle_size;j++)
{ System.out.print(previous[j]+"'"); }
System.out.println("\nvelocity :");
```

```
for(int j=0;j<Particle_size;j++)
{ System.out.print(velocity[j]+"'"); }
 }
public void Calculate_fitness
(Instance [] all_Instances,int total_instances,String [] Cats)
{
int selected_features=0;
 double alfa=0.85;
double beta=0.15;
for(int j=0; j< Particle_size;j++)
{
if(current[j]==1.0) { selected_features++;  }
  }
// calculate the accuracy of the features selected by
//current particle using e.g.
//KNN or SVM and use it to calculate the fitness of the particle
//fitness function here
Double fitness =
 alfa * accuracy + beta * ( ( (double) (Particle_size) -
(double) (selected_features) ) ) / (double) (Particle_size);
// if the new fitness is high than the previous one
// store it in fitness value parameter and
// store the current position of the particle
// in its previous position vector
if( fitness > fitness_value )
{
for(int j=0; j< Particle_size;j++)
{
previous[j] = current[j];
}
fitness_value=fitness;
}
}
public double get_fitness_value()
```

```
{ return fitness_value; }
// This method is used to update the particle's velocity
public void update_particle_velocity(Particle pg,double w)
{
Random generator = new Random();
int c1=2;
int c2=2;
for(int j=0; j< Particle_size;j++)
{ velocity[j]=(w*velocity[j])+
((double)c1*(generator.nextDouble()*1)*(previous[j]-current[j]))+
((double)c2*(generator.nextDouble()*1)*(pg.previous[j]-current[j]));
}}
// this method is used to update the particle's position
public void update_particle_position()
{
Random generator = new Random();
double sigmoid=0.0;
for(int j=0; j< Particle_size;j++)
{
sigmoid=(double)((double)1/(1+Math.exp(-1*velocity[j])));
if((generator.nextDouble()*1)<sigmoid)
current[j]=1.0;
else
current[j]=0.0;
}}
}  // end of particle class

import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.util.Iterator;
```

```
import java.util.TreeSet;
public class PSO {
// constructor
public PSO() { super(); }
public TreeSet<String>
Particle_Swarm_Optimization_Faetures_selection ( TreeSet < String >
features,Instance [] all_Instances,
int total_instances,String [] cat) throws IOException
{
int particle_size = features.size();
// Swarm Size
int swarm_size=30;
TreeSet<String> sub_set_of_features = new TreeSet<String>();
int bg_index=0;
// Create a population of particles with
//random positions and velocities
Particle [] p = new Particle[swarm_size];
for(int i=0;i<swarm_size;i++)
{
p[i]= new Particle(particle_size);
p[i].initilize();
}
double bestf=0.0;
int iterations=100;
int first_iteration=iterations;
double inertia =1.02;
int generation=1;
// start the iterations
do
// calculate the fitness of each particle in the swarm
for(int i=0;i<swarm_size;i++)
{
p[i].Calculate_fitness(all_Instances,total_instances,cat);
}
```

```
// in the first iteration
//We only compare the fitness because bg_index is empty
if(first_iteration==iterations)
{
bg_index=0;
bestf=p[0].get_fitness_value();
for(int i=0;i<swarm_size;i++)
{
if(p[i].get_fitness_value()>bestf)
{
bestf=p[i].get_fitness_value();
bg_index=i; // to catch the best one
}
}}
//Check for the best fitness value to find gbest
for(int i=0;i<swarm_size;i++)
{
if(p[i].get_fitness_value()>bestf)
{
bestf=p[i].get_fitness_value();
bg_index=i; // to catch the best one
}}
// update the velocity of each particle in the swarm
for(int i=0;i<swarm_size;i++)
{
p[i].update_particle_velocity(p[bg_index],inertia);
}
// update the position of each particle in the swarm
for(int i=0;i<swarm_size;i++)
{
p[i].update_particle_position();
}
// repeat until termination criteria is met
// i.e. a specific number of generations
```

```
generation++;
iterations--;
}
// end of generations' loop
while(iterations>0);
// Return the selected features only in sub_set_of_features TreeSet
for(int j=0;j<particle_size;j++)
{
int index =0;
if(p[bg_index].get_Xi(j)==1.0)
{
Iterator <String> s1 = features.iterator();
while(s1.hasNext())
{
String word = s1.next();
if(index==j)
{sub_set_of_features.add(word);}
index++;
}}}
return sub_set_of_features;
} // end of Particle_Swarm_Optimization_Faetures_selection method
} // end of PSO class

// This class is used to store text documents as
// TFIDF vectors of attributes.
public class Instance {
private double [] tfidf;
String instance_class;
// Constructor
public Instance(int f_size )
{
tfidf= new double[f_size];
for(int i=0;i<f_size;i++)
{  tfidf[i]=0.0; }
```

```
instance_class="";
}
public void set_class(String c) {instance_class=c;}
public String get_class() {return instance_class;}
public void set_tfidf(int index,double value) {tfidf[index]=value;}
public double get_tfidf_value(int index) {return tfidf[index];}
}
```

Bibliography

[1] A. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," in *LDV Forum-GLDV Journal for Computational Linguistics and Language Technology*, vol. 20, pp. 19–62, sn, 2005.

[2] M. P. Singh, *The practical handbook of internet computing*. Chapman & Hall/CRC, 2004.

[3] M. A. Wajeed and T. Adilakshmi, "Text Classification Using Machine Learning," *Journal of Theoretical and Applied Information Technology*, vol. 7, no. 2, pp. 119–123, 2009.

[4] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[5] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[6] F. Sebastiani, "Classification of text, automatic," *The Encyclopedia of Language and Linguistics*, vol. 14, pp. 457–462, 2006.

[7] H. Chen and T. K. Ho, "Evaluation of decision forests on text categorization," in *Proc. 7th SPIE Conference on Document Recognition and Retrieval*, pp. 191–199, Citeseer, 2000.

[8] C. Silva and B. Ribeiro, "The importance of stop word removal on recall values in text categorization," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 1661–1666, IEEE, 2003.

[9] F. Song, S. Liu, and J. Yang, "A comparative study on text representation schemes in text categorization," *Pattern Analysis & Applications*, vol. 8, no. 1, pp. 199–209, 2005.

[10] M. F. Caropreso, S. Matwin, and F. Sebastiani, "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization," *Text databases and document management: Theory and practice*, pp. 78–102, 2001.

[11] C. H. Koster and J. G. Beney, "Phrase-based document categorization revisited," in *Proceedings of the 2nd international workshop on Patent information retrieval*, pp. 49–56, ACM, 2009.

[12] W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowledge-Based Systems*, vol. 21, no. 8, pp. 879–886, 2008.

[13] M. C. Hung and Y. Don Lin, "An efficient fuzzy c-means clustering algorithm," in *Proceedings of IEEE International Conference on Data Mining*, pp. 225–232, 2001.

[14] D. Radošević, J. Dobša, D. Mladenić, Z. Stapić, and M. Novak, "Genre document classification using flexible length phrases," in *Proceedings of 17th International Conference on Information and Intelligent Systems*, 2006.

[15] N. Kapalavayi, S. J. Murthy, and G. Hu, "Document classification efficiency of phrase-based techniques," in *Proceedings of International Conference on Computer Systems and Applications*, pp. 174–178, IEEE, 2009.

[16] V. Nuipian, P. Meesad, and P. Boonrawd, "A comparison between keywords and key-phrases in text categorization using feature section technique," in *Proceedings of 9th International Conference on ICT and Knowledge Engineering*, pp. 156–160, IEEE, 2012.

[17] G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Learning rules for large vocabulary word sense disambiguation," in *Proceedings of International Joint Conference on Artificial Intelligence*, vol. 16, pp. 674–681, Lawrence Erlbaum Associates LTD, 1999.

[18] A. Montoyo, A. Suárez, G. Rigau, and M. Palomar, "Combining knowledge- and corpus-based word-sense-disambiguation methods," *Journal of Artificial Intelligence Research*, vol. 23, no. 1, pp. 299–330, 2005.

[19] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[20] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 160–167, ACM, 2000.

[21] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *The Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.

[22] Y. Yang, S. Slattery, and R. Ghani, "A study of approaches to hypertext categorization," *Journal of Intelligent Information Systems*, vol. 18, no. 2, pp. 219–241, 2002.

[23] T. M. Mitchell, *Machine learning*. McGraw Hill, 1997.

[24] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[25] Z. Markov and D. T. Larose, *Data mining the Web: uncovering patterns in Web content, structure, and usage*. Wiley-Interscience, 2007.

[26] E. T. Al-Shammari, "Improving arabic document categorization: Introducing local stem," in *Proceedings of 10th International Conference on Intelligent Systems Design and Applications*, pp. 385–390, IEEE, 2010.

[27] R. Duwairi, M. Al-Refai, and N. Khasawneh, "Stemming Versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization," in *Proceedings of 4th International Conference on Innovations in Information Technology*, pp. 446–450, IEEE, 2007.

[28] M. A. H. Omer and M. S. Long, "Stemming Algorithm to Classify Arabic Documents," 2010.

[29] L. Larkey, L. Ballesteros, and M. Connell, "Light stemming for arabic information retrieval," *Arabic computational morphology*, pp. 221–243, 2007.

[30] *Apache Lucene software, Text search engine library written in Java: http://lucene.apache.org/.*

[31] *Apache Lucene Java Core, Arabic normalizer and stemmer. Available at http://grepcode.com/snapshot/repo1.maven.org/maven2/org.apache.lucene/lucene-core/3.0.1.*

[32] *Apache Lucene, Arabic normalizer and stemmer. Available at http://grepcode.com/snapshot/repo1.maven.org /maven2/org.apache.lucene/luceneanalyzers/3.0.1.*

[33] Y. Li, D. F. Hsu, and S. M. Chung, "Combining multiple feature selection methods for text categorization by using rank-score characteristics," in *Proceedings of 21st International Conference on Tools with Artificial Intelligence*, pp. 508–517, IEEE, 2009.

[34] S. Doan and S. Horiguchi, "An efficient feature selection using multi-criteria in text categorization," in *Proceedings of Fourth International Conference on Hybrid Intelligent Systems*, pp. 86–91, IEEE, 2004.

[35] S. Li, R. Xia, C. Zong, and C.-R. Huang, "A framework of feature selection methods for text categorization," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pp. 692–700, Association for Computational Linguistics, 2009.

[36] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Machine Learning International Workshop then Conference*, pp. 412–420, Morgan Kaufmann Publishers, INC., 1997.

[37] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," in *ACM SIGIR Forum*, vol. 31, pp. 67–73, ACM, 1997.

[38] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.

[39] G. H. John, R. Kohavi, K. Pfleger, *et al.*, "Irrelevant features and the subset selection problem," in *Proceedings of the eleventh international conference on machine learning*, vol. 129, pp. 121–129, San Francisco, 1994.

[40] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[41] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving k-nearest-neighbor for classification," in *Proceedings of Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, vol. 1, pp. 679–683, IEEE, 2007.

[42] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[43] C.-W. Hsu, C.-C. Chang, C.-J. Lin, *et al.*, "A practical guide to support vector classification," 2003.

[44] S. Kotsiantis, "Supervised machine learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[45] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[46] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[47] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2, pp. 103–130, 1997.

[48] *Sample Applications using C5.0. Available at http://www.rulequest.com/see5-examples.html.*

[49] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[50] *Weka Software: http://www.cs.waikato.ac.nz/ml/weka/.*

[51] C. Blum and X. Li, "Swarm intelligence in optimization," *Swarm Intelligence*, pp. 43–85, 2008.

[52] V. Vassiliadis and G. Dounias, "NATURE–INSPIRED INTELLIGENCE: A REVIEW OF SELECTED METHODS AND APPLICATIONS," *International Journal on Artificial Intelligence Tools*, vol. 18, no. 04, pp. 487–516, 2009.

[53] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[54] B. Yang, Y. Chen, and Z. Zhao, "Survey on applications of particle swarm optimization in electric power systems," in *Proceedings of IEEE International Conference on Control and Automation*, pp. 481–486, IEEE, 2007.

[55] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of 1997 IEEE International Conference on Computational Cybernetics and Simulation*, vol. 5, pp. 4104–4108, IEEE, 1997.

[56] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of The 1998 World Congress on Computational Intelligence*, pp. 69–73, IEEE, 1998.

[57] Y. Shi *et al.*, "Particle swarm optimization: developments, applications and resources," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 81–86, IEEE, 2001.

[58] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization: An overview," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[59] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[60] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, IEEE, 1999.

[61] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 94–100, IEEE, 2001.

[62] P. R. Dian, M. S. Siti, and S. Y. Siti, "Particle Swarm Optimization: Technique, System and Challenges," *International Journal of Computer Applications*, vol. 14, no. 1, pp. 19–27, 2011.

[63] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1677–1681, IEEE, 2002.

[64] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.

[65] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1671–1676, IEEE, 2002.

[66] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

[67] L. Messerschmidt and A. P. Engelbrecht, "Learning to play games using a PSO-based competitive learning approach," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 280–288, 2004.

[68] H. Zhenya, W. Chengjian, Y. Luxi, G. Xiqi, Y. Susu, R. C. Eberhart, and Y. Shi, "Extracting rules from fuzzy neural network by particle swarm optimisation," in *Proceedings of IEEE World Congress on Computational Intelligence*, pp. 74–77, IEEE, 1998.

[69] C. Zhang and H. Shao, "An ANN's evolved by a new evolutionary system and its application," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 4, pp. 3562–3563, IEEE, 2000.

[70] A. Abido, "Particle swarm optimization for multimachine power system stabilizer design," in *Power Engineering Society Summer Meeting*, vol. 3, pp. 1346–1351, IEEE, 2001.

[71] A. A. Esmin, G. Lambert-Torres, and A. C. Zambroni de Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 859–866, 2005.

[72] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions," *IEEE Transactions on Power systems*, vol. 20, no. 1, pp. 34–42, 2005.

[73] X. Zhang, W. Hu, S. Maybank, X. Li, and M. Zhu, "Sequential particle swarm optimization for visual tracking," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[74] S. Cagnoni, M. Mordonini, and J. Sartori, "Particle swarm optimization for object detection and segmentation," *Applications of Evolutionary Computing*, pp. 241–250, 2007.

[75] R. M. Ramadan and R. F. Abdel-Kader, "Face recognition using particle swarm optimization-based selected features," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 2, pp. 51–65, 2009.

[76] J. Pugh and A. Martinoli, "Inspiring and modeling multi-robot search with particle swarm optimization," in *Proceedings of IEEE Swarm Intelligence Symposium*, pp. 332–339, IEEE, 2007.

[77] S. Doctor, G. K. Venayagamoorthy, and V. G. Gudise, "Optimal PSO for collective robotic search applications," in *Proceedings of Congress on Evolutionary Computation*, vol. 2, pp. 1390–1395, IEEE, 2004.

[78] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[79] W. Liu and D. Zhang, "Feature Subset Selection Based on Improved Discrete Particle Swarm and Support Vector Machine Algorithm," in *Proceedings of International Conference on Information Engineering and Computer Science*, pp. 1–4, IEEE, 2009.

[80] L.-Y. Chuang, C.-H. Yang, and J.-C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection," *Applied Soft Computing*, vol. 11, no. 1, pp. 239–248, 2011.

[81] B. Wei, Q. Peng, C. Li, and X. Kang, "A hybrid of binary Particle Swarm Optimization and estimation distribution algorithm for feature selection," in *Proceedings of Sixth International Conference on Natural Computation*, vol. 5, pp. 2510–2514, IEEE, 2010.

[82] A. Li and B. Wang, "Feature Subset Selection Based on Binary Particle Swarm Optimization and Overlap Information Entropy," in *Proceedings of International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, IEEE, 2009.

[83] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.

[84] D. Said, N. M. Wanas, N. M. Darwish, and N. Hegazy, "A study of text preprocessing tools for arabic text categorization," in *Proceedings of The Second International Conference on Arabic Language*, pp. 230–236, 2009.

[85] R. Al-Shalabi and R. Obeidat, "Improving KNN Arabic Text Classification with N-Grams Based Document Indexing," in *Proceedings of the Sixth International Conference on Informatics and Systems, Cairo, Egypt*, 2008.

[86] F. Thabtah *et al.*, "VSMs with K-Nearest Neighbour to Categorise Arabic Text Data," 2008.

[87] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. Khorsheed, and A. Al-Rajeh, "Automatic Arabic text classification," 2008.

[88] A. El-Halees, "Arabic text classification using maximum entropy," *The Islamic University Journal*, vol. 15, no. 1, pp. 157–167, 2007.

[89] M. El Kourdi, A. Bensaid, and T.-e. Rachidi, "Automatic arabic document categorization based on the naïve bayes algorithm," in *Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages*, pp. 51–58, 2004.

[90] A. Moh'd A MESLEH, "Chi square feature extraction based SVMs Arabic language text categorization system," *Journal of Computer Science*, vol. 3, no. 6, pp. 430–435, 2007.

[91] T. Rachidi, O. Iraqi, M. Bouzoubaa, A. Khattab, M. Kourdi, A. Zahi, and A. Bensaid, "Barq: distributed multilingual internet search engine with focus on arabic language," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 428–435, IEEE, 2003.

[92] F. Thabtah, M. Eljinini, M. Zamzeer, and W. Hadi, "Naïve Bayesian based on Chi Square to Categorize Arabic Data," in *proceedings of The 11th International Business Information Management Association Conference (IBIMA) Conference on Innovation and Knowledge Management in Twin Track Economies, Cairo, Egypt*, pp. 930–935, 2009.

[93] R. Duwairi, "Arabic text categorization," *the international Arab Journal of information Technology*, vol. 7, 2007.

[94] L. Khreisat, "Arabic text classification using N-gram frequency statistics a comparative study," in *Conference on Data Mining— DMIN*, vol. 6, pp. 78–82, 2006.

[95] A. Mesleh, "Support vector machines based arabic language text classification system: feature selection comparative study," in *Proceedings of the 12th WSEAS international Conference on Applied Mathematics*, pp. 228–233, World Scientific and Engineering Academy and Society (WSEAS), 2007.

[96] S. M. Al-Saleem, "Associative Classification to Categorize Arabic Data Sets," *International Journal of ACM Jordan*, vol. 1, pp. 118–127, 2010.

[97] G. Kanaan, R. Al-Shalabi, and A. Al-Akhras, "KNN Arabic text categorization using IG feature selection," in *Proceedings of The 4th International Multiconference on Computer Science and Information Technology*, vol. 4, pp. 5–7, 2006.

[98] L. Khreisat, "A machine learning approach for arabic text classification using N-gram frequency statistics," *Journal of Informetrics*, vol. 3, no. 1, pp. 72–77, 2009.

[99] B. M. Zahran and G. Kanaan, "Text feature selection using particle swarm optimization algorithm," *World Applied Sciences Journal*, vol. 7, pp. 69–74, 2009.

[100] *Machine Learning Open Source Software. Available at http://jmlr.org/mloss/.*

[101] M. K. Saad and W. Ashour, "Arabic Text Classification Using Decision Trees," in *Proceedings of the 12th international workshop on computer science and information technologies CSIT*, pp. 75–79, 2010.

[102] I. Hmeidi, B. Hawashin, and E. El-Qawasmeh, "Performance of KNN and SVM classifiers on full word Arabic articles," *Advanced Engineering Informatics*, vol. 22, no. 1, pp. 106–111, 2008.

[103] W. M. Hadi, M. Salam, and J. A. Al-Widian, "Performance of NB and SVM classifiers in Islamic Arabic data," in *Proceedings of the 1st International Conference on Intelligent Semantic Web-Services and Applications*, ACM, 2010.

[104] S. Raheel and J. Dichy, "An Empirical Study on the Features Type Effect on the Automatic Classification of Arabic Documents," *Computational Linguistics and Intelligent Text Processing*, pp. 673–686, 2010.

[105] S. Alsaleem, "Automated Arabic Text Categorization Using SVM and NB," *International Arab Journal of e-Technology*, vol. 2, no. 2, pp. 124–128, 2011.

[106] M. Al-diabat, "Arabic Text Categorization Using Classification Rule Mining," *Applied Mathematical Sciences*, vol. 6, no. 81, pp. 4033–4046, 2012.

[107] F. Harrag, E. El-Qawasmeh, and P. Pichappan, "Improving Arabic text categorization using decision trees," in *Proceedings of The First International Conference on Networked Digital Technologies*, pp. 110–115, IEEE, 2009.

[108] F. Thabtah, O. Gharaibeh, and R. Al-Zubaidy, "Arabic Text Mining Using Rule Based Classification," *Journal of Information & Knowledge Management*, vol. 11, no. 01, 2012.

[109] A. M. Mesleh, "Feature sub-set selection metrics for arabic text classification," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1922–1929, 2011.

[110] Z. Elberrichi and K. Abidi, "Arabic Text Categorization: A comparative Study of Different Representation Modes," *International Arab Journal of Information Technology*, vol. 9, no. 5, 2012.

[111] A. Mesleh and G. Kanaan, "Support vector machine text classification system: Using Ant Colony Optimization based feature subset selection," in *Proceedings of International Conference on Computer Engineering & Systems*, pp. 143–148, IEEE, 2008.

[112] *Arabic datasets. Available at http://is.gd/arabdata.*

[113] *Arabic datasets. Available at http://sites.google.com/site/mouradabbas9/corpora.*

[114] M. Abbas and K. Smaïli, "Comparison of topic identification methods for arabic language," in *International Conference on Recent Advances in Natural Language Processing*, vol. 14, 2005.

[115] M. Abbas, K. Smaili, and D. Berkani, "Comparing TR-Classifier and KNN by using Reduced Sizes of Vocabularies," 2009.

[116] *Arabic dataset. Available at http://filebox.vt.edu/users/dsaid/Alj-News.tar.gz.*

[117] Y. El-Manzalawy and V. Honavar, "WLSVM: integrating libsvm into weka environment," *Software available at http://www. cs. iastate. edu/yasser/wlsvm*, 2005.

[118] M. Tayli and A. I. Al-Salamah, "Building bilingual microcomputer systems," *Communications of the ACM*, vol. 33, no. 5, pp. 495–504, 1990.

[119] I. A. El-Khair, "Effects of stop words elimination for Arabic information retrieval: a comparative study," *International Journal of Computing & Information Sciences*, vol. 4, no. 3, pp. 119–133, 2006.

[120] B. Al-Shargabi, W. Al-Romimah, and F. Olayah, "A comparative study for Arabic text classification algorithms based on stop words elimination," in *Proceedings of the 2011 International Conference on Intelligent Semantic Web-Services and Applications*, ACM, 2011.

[121] M. A. Aabed, S. M. Awaideh, A.-R. Elshafei, and A. A. Gutub, "Arabic diacritics based steganography," in *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, pp. 756–759, IEEE, 2007.

[122] I. Zitouni, J. S. Sorensen, and R. Sarikaya, "Maximum entropy based restoration of arabic diacritics," in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pp. 577–584, Association for Computational Linguistics, 2006.

[123] J. Kennedy, "Bare bones particle swarms," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pp. 80–87, IEEE, 2003.

[124] W. Wright, *A Grammar of the Arabic Language*. Librairie du Liban, 1974.

[125] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[126] E. T. Al-Shammari and J. Lin, "Towards an error-free Arabic stemming," in *Proceedings of the 2nd ACM workshop on Improving non english web searching*, pp. 9–16, ACM, 2008.

[127] S. Khoja, "APT: Arabic part-of-speech tagger," in *Proceedings of the Student Workshop at NAACL*, pp. 20–25, 2001.

[128] S. Khoja, R. Garside, and G. Knowles, "A tagset for the morphosyntactic tagging of Arabic," *Proceedings of the Corpus Linguistics. Lancaster University (UK)*, vol. 13, 2001.

[129] M. K. Saad and W. Ashour, "Arabic Morphological Tools for Text Mining," 2010.

[130] A. F. Nwesri, S. M. Tahaghoghi, and F. Scholer, "Capturing out-of-vocabulary words in Arabic text," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 258–266, Association for Computational Linguistics, 2006.

[131] M. Aljlayl and O. Frieder, "On arabic search: improving the retrieval effectiveness via a light stemming approach," in *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 340–347, ACM, 2002.

[132] A. Mesleh, *Support Vector Machine Text Classifier for Arabic Articles: Ant Colony Optimization-Based feature selection*. 2008.

[133] R. Gutierrez-Osuna, "Introduction to pattern analysis," *Lecture Notes, Texas A&M University*, 2005.

[134] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[135] K. Baek, B. A. Draper, J. R. Beveridge, and K. She, "PCA vs. ICA: A comparison on the FERET data set," in *Joint Conference on Information Sciences, Durham, NC*, pp. 824–827, 2002.

[136] K. Torkkola, "Discriminative features for text document classification," *Pattern Analysis & Applications*, vol. 6, no. 4, pp. 301–308, 2004.

[137] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[138] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[139] T. Liu, Z. Chen, B. Zhang, W.-y. Ma, and G. Wu, "Improving text classification using local latent semantic indexing," in *Proceedings of Fourth IEEE International Conference on Data Mining*, pp. 162–169, IEEE, 2004.

[140] L. Chen, N. Tokuda, and A. Nagai, "A new differential LSI space-based probabilistic document classifier," *Information Processing Letters*, vol. 88, no. 5, pp. 203–212, 2003.

[141] S. Zelikovitz and H. Hirsh, "Using LSI for text classification in the presence of background text," in *Proceedings of the tenth international conference on Information and knowledge management*, pp. 113–118, ACM, 2001.

[142] S. Goswami and M. S. Shishodia, "A Fuzzy Based Approach to Text Mining and Document Clustering," *arXiv preprint arXiv:1306.4633*, 2013.

[143] H. K. Chantar and D. W. Corne, "Feature subset selection for arabic document categorization using BPSO-KNN," in *Nature and Biologically Inspired Computing, 2011 Third World Congress on*, pp. 546–551, IEEE, 2011.

[144] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information processing letters*, vol. 85, no. 6, pp. 317–325, 2003.

[145] E. Ozcan and C. K. Mohan, "Analysis of a simple particle swarm optimization system," *Intelligent engineering systems through artificial neural networks*, vol. 8, pp. 253–258, 1998.